

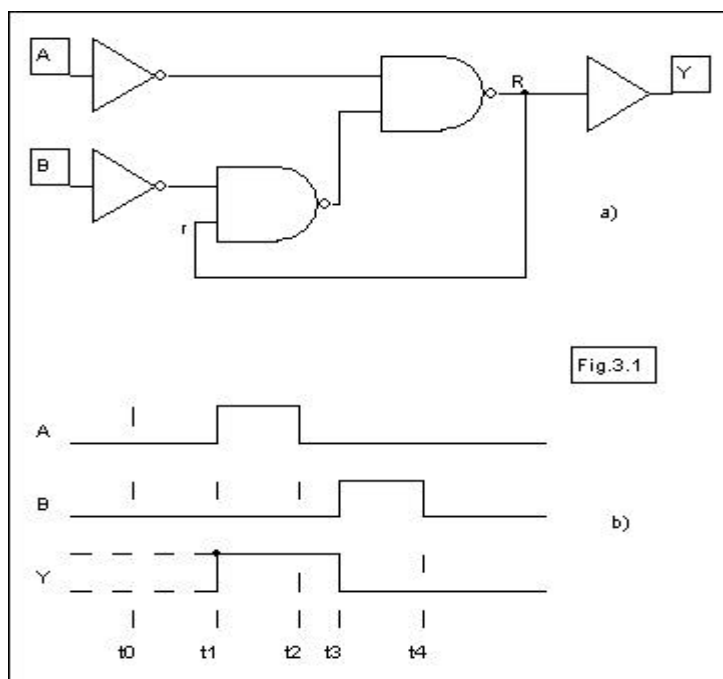
## CAPITOLO III

### LE RETI SEQUENZIALI

#### 3.1 INTRODUZIONE

Le reti sequenziali sono reti logiche nelle quali il valore dell'uscita  $Y$  ad un certo istante  $t$  è funzione sia del valore assunto dalle variabili di entrata in quello stesso istante, sia dei valori assunti dalle medesime variabili prima di  $t$ .

Queste reti hanno quindi memoria.



Consideriamo, per esempio, la rete di **Fig. 3.1a**. Sia 00 (cioè,  $A=0$ ,  $B=0$ ) lo stato di entrata a  $t=0$ ; è facile verificare che lo stato di uscita può essere indifferentemente 0 oppure 1. Quindi, la sola conoscenza dello stato di entrata non permette di conoscere quale sarà lo stato di uscita corrispondente.

Se nella stessa rete lo stato di entrata all'istante  $t=t_1$  diventa 10, allora è  $Y=1$ . Se, a  $t=t_2$ ,  $A$  torna a 0,  $Y$

rimane a 1, come è facile verificare. Quindi la sequenza di stati di entrata

00 --> 10 --> 00

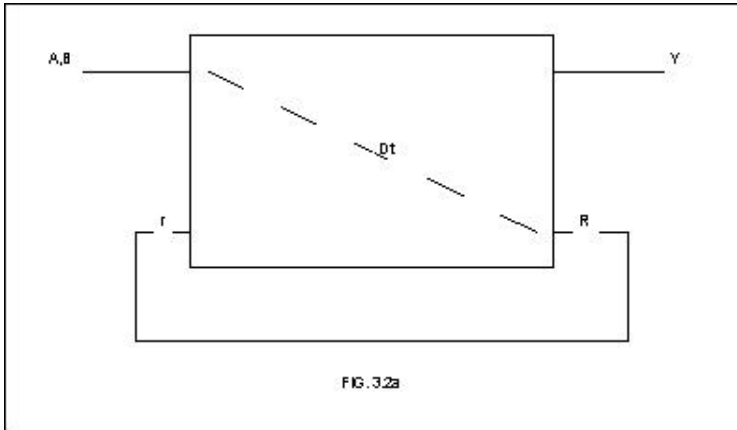
vede  $Y=1$  come stato di uscita finale.

Se invece consideriamo la sequenza di entrata

00 --> 01 --> 00

lo stato di uscita finale è  $Y=0$ .

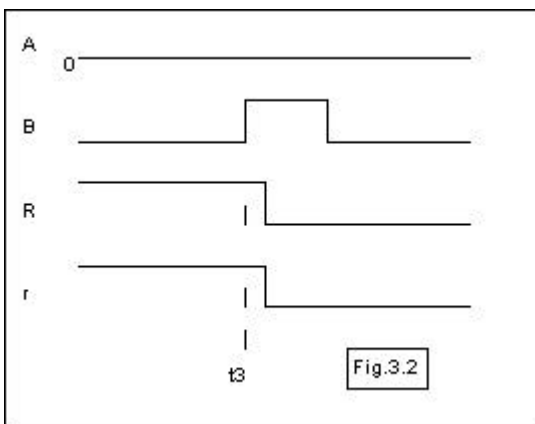
Quanto ora detto è descritto nel diagramma temporale di **Fig. 3.1b**. Quindi, per conoscere lo stato



di uscita corrispondente allo stato attuale di entrata, occorre in generale conoscere la sequenza di transizioni attraverso cui l'entrata ha raggiunto lo stato attuale: il circuito "ricorda" la sequenza.

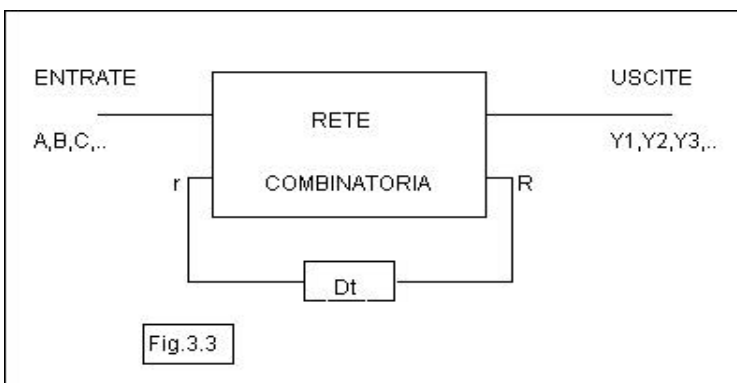
Osservando la rete di **Fig. 3.1a**, si nota la presenza di un ingresso alla

rete il cui stato logico non è definito dall'esterno bensì internamente alla rete, tramite un collegamento di reazione ingresso-uscita i cui estremi sono identificati dalle lettere R e r: questo collegamento tiene "informato" l'ingresso circa lo stato dell'uscita "interna" R, e quindi circa la storia passata del circuito.



Tuttavia, la presenza di una o più spire di reazione in una rete logica non deve comunque far concludere che essa è sequenziale: si possono infatti dare esempi in cui questo non è vero. La condizione necessaria e sufficiente perché una rete sia di natura sequenziale è che per essa esista almeno uno stato di entrata cui corrispondono due o più stati di uscita in cui la rete può permanere indefinitamente (stati stabili). In una

rete combinatoria ogni stato di entrata individua univocamente uno stato di uscita; in una rete sequenziale, invece, il numero di stati stabili di uscita è maggiore del numero di stati di entrata.



In effetti, la **Fig. 3.1b** non dà una descrizione temporalmente esatta di quanto avviene in una rete sequenziale reale, perché non tiene conto del ritardo di propagazione. Se indichiamo con  $\delta t$  il ritardo di propagazione tra l'ingresso B e l'uscita R, **Fig. 3.2a**, la situazione reale è illustrata in **Fig. 3.2**: quando, all'istante  $t_3$ , B è transitato da 0 a 1, lo stato logico  $r$  all'estremità di ingresso della spira di reazione è ancora a 1. Pertanto, l'uscita R (e anche Y) è destinata ad andare in uno stato che dipende sia dal *nuovo stato di entrata*, 01, sia dal *vecchio stato di uscita*, 1, (rappresentato in ingresso da  $r$ ). Il ritardo  $\delta t$  costituisce quindi la "memoria" della rete.

Ai fini dell'analisi del comportamento della rete, è conveniente immaginare il ritardo  $\delta t$  concentrato in un punto della spira di reazione, nel tratto che collega l'uscita R con l'ingresso  $r$ , come mostrato in **Fig. 3.3**. La rete sequenziale appare quindi schematizzata come una rete combinatoria con risposta istantanea, più un elemento di ritardo. (Si noti che in una rete sequenziale possono esserci più spire di reazione, ciascuna col proprio ritardo).

Il valore logico di R si chiama "*stato interno seguente*" della rete. Questo, in generale, differisce dall'uscita Y. Il valore logico di  $r$  si chiama invece "*stato interno presente*".

Dalla **Fig. 3.3** appare che, poiché la rete combinatoria ha risposta istantanea, all'atto di una transizione di entrata cambia istantaneamente lo stato seguente R, mentre lo stato presente  $r$  resta invariato. Al momento della transizione, lo stato seguente R può differire dallo stato presente  $r$  e si dice allora che la rete è in uno *stato instabile*. In ogni caso, è evidente che,  $\delta t$  secondi dopo la transizione,  $r$  e R devono coincidere. Se l'attuale stato di entrata ed il nuovo stato interno presente non comportano un ulteriore cambiamento dello stato seguente, si dice che la rete è in uno *stato stabile*. Può accadere che occorran più transizioni di stato interno prima che la rete raggiunga uno stato stabile; se questo numero non è finito, si dice che all'attuale stato di entrata non corrispondono stati stabili della rete.

La **Fig. 3.3** mette in evidenza che, a partire dall'istante in cui si ha una transizione nello stato di entrata, la spira di reazione resta in effetti "aperta" per  $\delta t$  secondi: in questo tempo il circuito ha un comportamento puramente combinatorio, e lo stato seguente R è esprimibile come una funzione booleana dello stato di entrata e dello stato interno presente  $r$ ; questa funzione può quindi essere analizzata con i metodi già studiati. Ciò è illustrato nel paragrafo seguente.

Infine, da quanto detto è evidente che lo stato di uscita della rete cambia al momento in cui cambia lo

-3.4-

stato di entrata: una rete sequenziale siffatta è chiamata asincrona, per distinguerla da una rete sequenziale sincrona che sarà introdotta più avanti.

### 3.2 ANALISI DI UNA RETE SEQUENZIALE ASINCRONA

L'analisi consiste nel determinare la sequenza di stati di uscita per una assegnata sequenza di stati di entrata. Per questo, occorre innanzi tutto ricercare gli stati stabili della rete. Questa ricerca viene effettuata in due passi:

- 1- si determina la risposta interna istantanea della rete

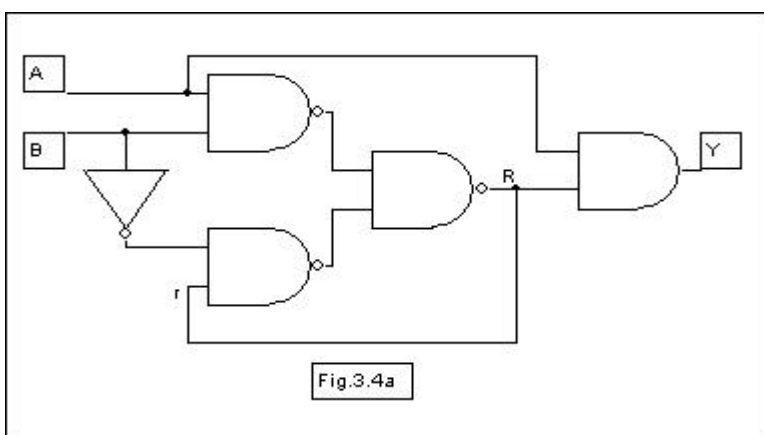
$$R=f(A,B,\dots,r_1,r_2,\dots) \text{ e la si tabula;}$$

sulla tavola si studia l'evoluzione della rete dopo  $\Delta t$ , alla ricerca degli stati stabili.

- 2- Successivamente, si determina la sequenza di uscita

$$Y=f(A,B,\dots,R_1,R_2,\dots) \text{ e la si tabula.}$$

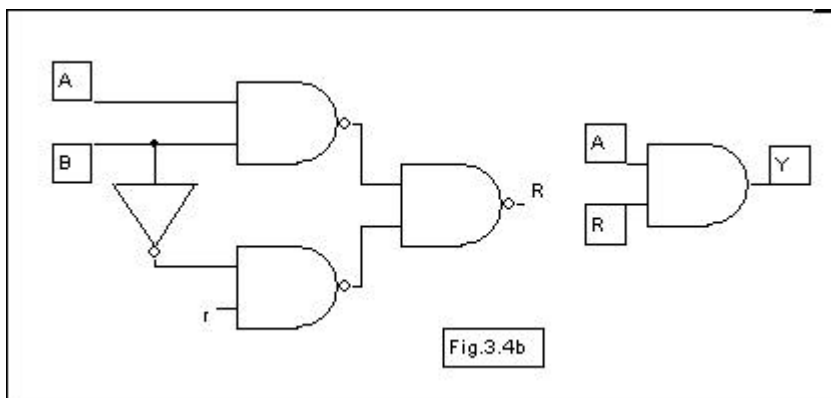
Su questa tavola si studia l'evoluzione dell'uscita.



Per illustrare il procedimento di analisi, riferiamoci all'esempio di **Fig. 3.4a**. In **Fig. 3.4b** la rete è ridisegnata a spirata aperta, cioè nella situazione in cui essa si trova per  $\delta t$  secondi a partire da una transizione di entrata.

Dalla **Fig. 3.4b** possiamo determinare la "funzione dello stato seguente", cioè la funzione booleana di R:

$$R = AB + \bar{B}r$$



Questa funzione è tabulata nella "tavola delle transizioni", **Fig. 3.5a**, che è strutturata come una mappa di Karnaugh, con una opportuna disposizione delle variabili: precisamente, si pone

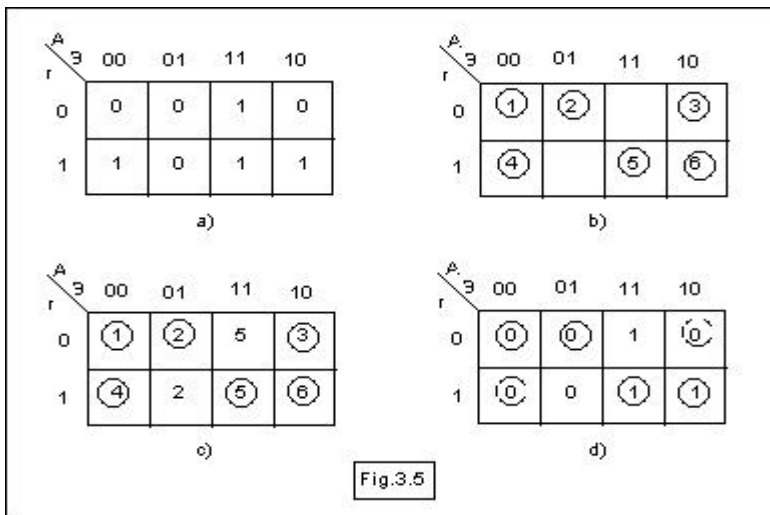
come indice di riga lo stato presente, e come indice di colonna lo stato di entrata.

Come già detto, gli stati stabili della rete sono quelli in cui  $R = r$ : numerandoli arbitrariamente e cerciandoli, si ha la tavola di **Fig. 3.5b**. Le caselle vuote corrispondono a stati instabili.

La tavola permette di studiare il comportamento della rete per le possibili transizioni di entrata, come ora si fa vedere.

Supponiamo che la rete si trovi inizialmente nello stato stabile 1, cioè sia  $A=B=r=0$ . Supponiamo poi che lo stato di entrata effettui la transizione

00 -> 01

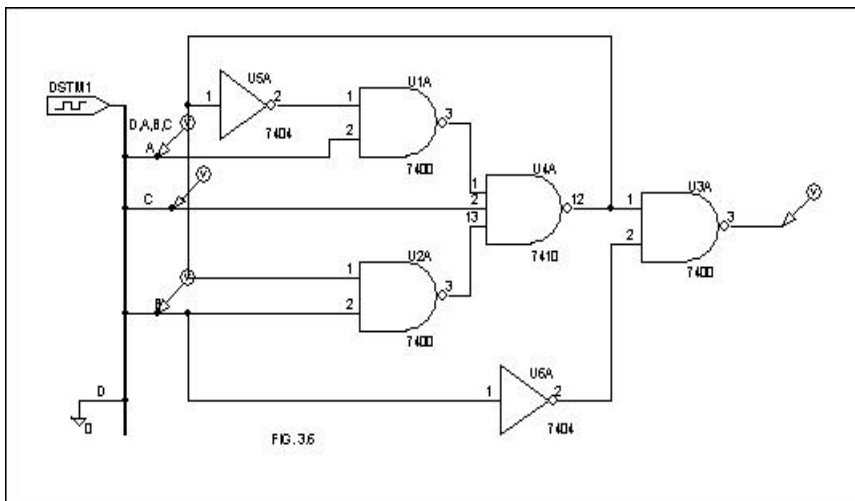


Poiché immediatamente dopo la transizione lo stato presente  $r$  non può cambiare (potrà cambiare dopo  $\delta t$  secondi, **Fig. 3.3**), ci si sposta orizzontalmente, sulla tavola, nella casella corrispondente al nuovo stato di entrata, cioè nello stato stabile 2.

Consideriamo ora la transizione

00 -> 10

La rete raggiunge direttamente lo stato stabile 3, ove si ferma.



Consideriamo la transizione

10 -> 11

Lo stato immediatamente raggiunto dalla rete (terza casella della prima riga) non è stabile, poiché qui è  $R=1$ , mentre lo stato

presente è  $r=0$ . In questo stato la rete può restare solo per  $\delta t$  secondi, dopo di che  $r$  deve cambiare per diventare uguale ad  $R$ : questo significa che si avrà una transizione verticale nella tavola, verso la riga in cui  $r=1$ . Qui la rete trova lo stato stabile 5, in cui essa si ferma.

Possiamo ora attribuire un numero anche a ciascuno stato instabile: è naturale assumere tale numero uguale a quello dello stato stabile che la rete è destinata a raggiungere.

L'analisi del comportamento della rete può ora essere facilmente completato, ottenendo la "tavola di flusso", Fig. 3.5c.

Dalla rete di Fig. 3.4b si può determinare l'equazione dell'uscita

$$Y = AR$$

Si noti che  $Y$  viene fatto dipendere da  $R$ , non da  $r$ : ciò perché interessa lo stato di uscita *dopo* che l'entrata ha cambiato stato. [Tuttavia, se questo è corretto per gli stati stabili, è in effetti arbitrario

far dipendere il valore dell'uscita negli stati instabili dallo stato seguente  $R$ ].

La tavola dell'uscita è in Fig. 3.5d.

Come secondo esempio, consideriamo la rete di Fig. 3.6, la cui funzione di stato seguente è

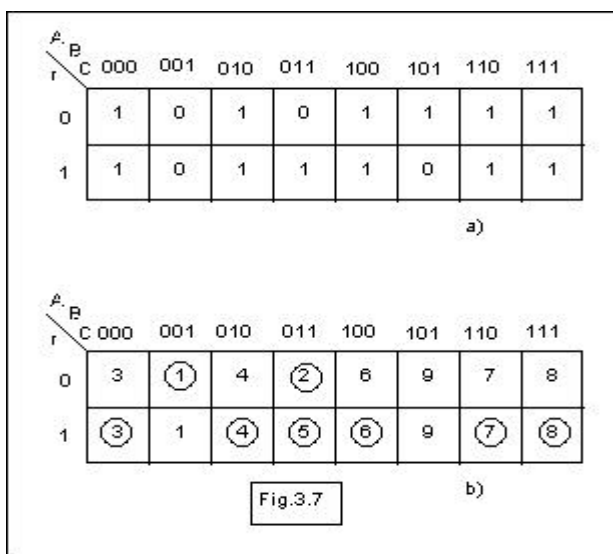
$$R = A\bar{r} + Br + \bar{C}$$

Le Fig. 3.7a e 3.7b mostrano la tavola delle transizioni e quella di flusso.

Appare subito la mancanza di stati stabili in

corrispondenza dello stato di entrata 101. Pertanto, quando per esempio l'entrata effettua la transizione  $100 \rightarrow 101$ , dallo stato stabile 6 si passa in uno stato in cui è  $R=0, r=1$ , cioè uno stato instabile: pertanto, dopo  $\delta t$  secondi lo stato presente è costretto a cambiare, diventando 0. La rete si è ora spostata nella casella  $r=0, R=1$ , che è ancora instabile (i cambiamenti di stato interno sono vincolati a restare nella stessa colonna, finché non cambia l'entrata): è chiaro che lo stato interno della rete continuerà ad oscillare fra i due stati instabili, fino a quando l'entrata non cambia.

La funzione di uscita è



-3.8-

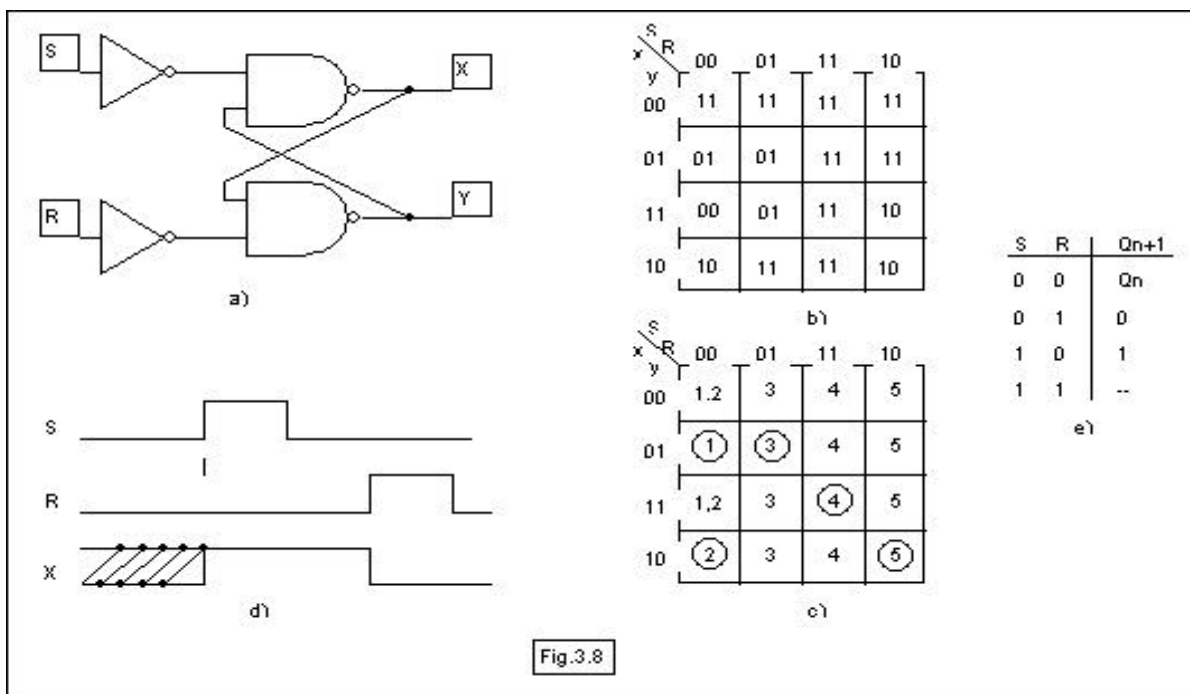
$$Y = \overline{\overline{BR}} = B + \overline{R}$$

la quale mette in evidenza che l'oscillazione dello stato interno sarà visibile anche in uscita.



### 3.3 IL FLIP-FLOP SR

Riconsideriamo la rete dell'esempio di **Fig. 3.1a**, ridisegnandola nella variante di **Fig. 3.8a**, nota



come flip-flop (FF) Set-Reset (SR).

Si noti che la rete ha due spire di reazione, e che gli stati seguenti coincidono con le uscite. Usiamo lo stesso simbolo per identificare lo stato seguente e l'uscita.

Le funzioni di stato seguente sono

$$X = S + \bar{y} \qquad Y = R + \bar{x}$$

La tavola delle transizioni è in **Fig. 3.8b**: in ciascuna casella, il primo valore è X, il secondo è Y. La tavola di flusso è in **Fig. 3.8c**.

Analizziamo le possibili transizioni di entrata, partendo dallo stato stabile 4, corrispondente allo stato di entrata 11, e consideriamo dapprima la transizione 11 -> 00.

La tavola delle transizioni sembra indicare una condizione di oscillazione della rete. In realtà, questo non si verifica mai, per due motivi. Il primo è che il cambiamento contemporaneo delle due entrate (che in generale provengono da sorgenti indipendenti) è un evento praticamente impossibile, a causa dei differenti ritardi di propagazione dei vari percorsi e della loro dipendenza da fattori incontrollabili (invecchiamento, variazioni di temperatura, ecc.). Pertanto, è verosimile che l'entrata

-3.10-

effettui la transizione indicata non direttamente, ma seguendo il percorso

$$11 \rightarrow 10 \rightarrow 00$$

oppure il percorso

$$11 \rightarrow 01 \rightarrow 00$$

Nel primo caso, le transizioni di stato interno saranno (per comodità, si indica in grassetto corsivo lo stato instabile)

$$4 \rightarrow \mathbf{5} \rightarrow 5 \rightarrow 2$$

cioè la rete si ferma in 2. Nel secondo caso si avrà

$$4 \rightarrow \mathbf{3} \rightarrow 3 \rightarrow 1$$

e la rete si ferma in 1.

Il secondo motivo è che anche se le due entrate cambiano contemporaneamente, non sarà così verosimilmente per x e y, perché i segnali attraversano porte diverse; i percorsi possibili sono

$$4 \rightarrow 1,2 \rightarrow 2$$
$$4 \rightarrow 1,2 \rightarrow 1$$

Non essendo prevedibile, in entrambi i casi, quale sarà di volta in volta il reale percorso, non è prevedibile lo stato di uscita.

Se lo stato di entrata diventa 01 (non importa da dove provenga), lo stato seguente diventa 01.

Analogamente, se l'entrata transita in 10 (non importa da dove provenga), lo stato seguente diventa 10.

Consideriamo, ora, la sequenza di transizioni di entrata

$$00 \rightarrow 10 \rightarrow 00$$

La corrispondente sequenza di stati di uscita è

$$** \rightarrow 10 \rightarrow 10$$

(\*\* indica uno stato indefinito).

Analogamente, considerando la sequenza di entrata

-----

00 -> 01 -> 00

si ha in uscita la sequenza

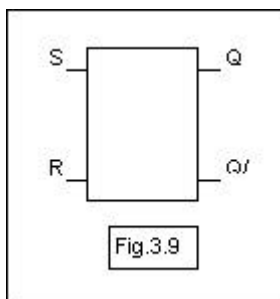
\*\* -> 01 -> 01

Poiché le due uscite sono una il complemento dell'altra, possiamo caratterizzare lo stato della rete con la sola uscita X. Possiamo, allora, concludere quanto segue:

- se l'ingresso S viene posto temporaneamente nello stato 1 (mentre R è a 0), l'uscita della rete si porta definitivamente nello stato 1. Detto in altri termini, si è "scritto" un 1 nel FF, e questo 1 resta memorizzato anche dopo che l'ingresso di Set torna a 0.
- se l'ingresso R viene posto temporaneamente nello stato 1 (mentre S=0), il FF si porta definitivamente nello stato 0; cioè, l'1 scritto in precedenza viene cancellato.
- In ogni caso, quando l'ingresso della rete torna nello stato 00, lo stato di uscita resta invariato, cioè lo stato seguente è identico allo stato presente.

I diagrammi temporali di **Fig. 3.8d** descrivono in modo pittorico quanto ora detto.

Lo stato di entrata 00 può essere inteso come uno stato di riposo.



Si esprime quanto sopra illustrato dicendo che nel FF è possibile immagazzinare un bit di informazione. Il FF è un elemento di memoria.

Questa è l'applicazione peculiare del FF: pertanto, da questo punto di vista appare evidente che lo stato di entrata 11 non può essere preso in considerazione, poiché non ha senso scrivere e cancellare contemporaneamente un dato nella memoria.

Usualmente, le uscite del FF sono chiamate Q e  $\bar{Q}$  (indicato come Q/ in tutte le figure), rispettivamente. Si indica con  $Q^n$  lo stato presente del FF, e con  $Q^{n+1}$  lo stato seguente. La descrizione precedente del comportamento del FF viene normalmente riassunta nella "tavola caratteristica" di **Fig. 3.8e**.

Il simbolo circuitale del FF è in **Fig. 3.9**.

### 3.4 RETI SEQUENZIALI ASINCRONE E SINCRONE

Nella illustrazione finora fatta di una rete sequenziale, vedi Fig. 3.3, la spira di reazione è *sempre chiusa*: pertanto, non appena cambia lo stato di entrata, cambiano lo stato seguente  $R$  e l'uscita  $Y$ ; dopo  $\delta t$  sec cambia anche lo stato interno presente  $r$ . Il fatto che la spira sia sempre chiusa richiede che *lo stato logico di  $R$  e  $r$  sia lo stesso* perché la rete raggiunga uno stato stabile. Questo fa sì che i cambiamenti di stato interno possano ripetersi più volte, cioè la rete può attraversare uno o più stati instabili prima di raggiungere uno stato stabile.

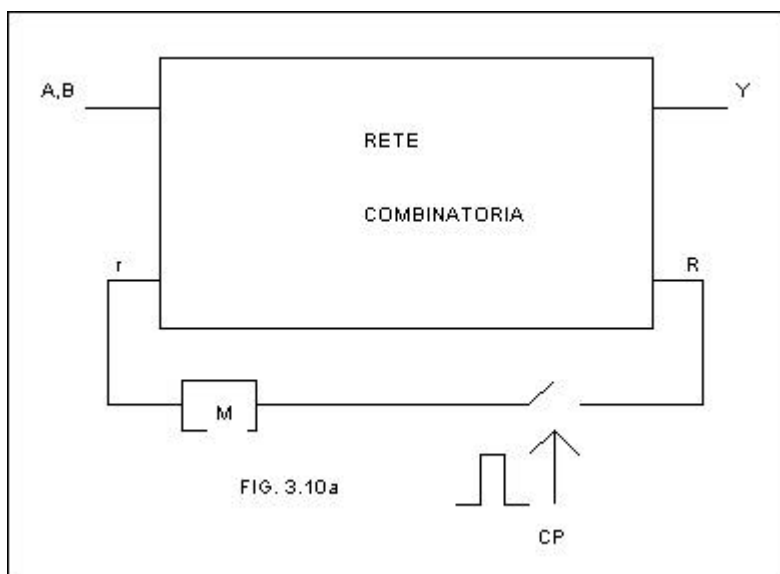
In linea di principio, nella rete di fig. 3.3 lo stato di entrata può cambiare in qualunque momento, e si dice che la rete sequenziale funziona in modo *asincrono*. Tuttavia, il funzionamento asincrono deve tener conto dell'intervallo di tempo, variabile, necessario per raggiungere uno stato stabile: cioè, affinché la risposta sia non equivoca, occorre che la rete abbia raggiunto lo stato stabile relativo alla  $n$ -ma transizione di entrata, prima di poter effettuare la  $n+1$ -ma transizione di entrata.

Inoltre, a causa della inevitabile differenza nei ritardi di propagazione dei segnali lungo diversi percorsi all'interno della rete, è praticamente impossibile che il cambiamento di stato di due o più variabili, pur generato nello stesso istante, arrivi contemporaneamente in un altro punto della rete seguendo percorsi diversi; in più, la differenza nei ritardi non è costante, a causa delle tolleranze, della temperatura, ecc. Si è già visto come differenze nei ritardi di propagazione possano dar luogo ad anomalie, sia pure transitorie, nelle reti combinatorie (alee). Inconvenienti possono aversi per loro causa anche nelle reti sequenziali asincrone perché, essendo aleatorio l'ordine di arrivo dei segnali nei vari punti della rete, l'uscita può portarsi in uno stato errato. Poiché una rete sequenziale ha memoria, lo stato errato può essere memorizzato stabilmente. La comprensione di questi problemi (noti come corse, alee statiche, alee essenziali) è ovviamente cruciale per il corretto funzionamento di una rete asincrona, ma non possiamo qui occuparcene. Si rimanda alla letteratura.

Per evitare questi inconvenienti, nelle reti asincrone è necessario imporre la condizione che ogni qualvolta si provoca un cambiamento di stato *debba cambiare una sola variabile* (sia di entrata sia interna).

I suddetti inconvenienti possono essere evitati, senza dover imporre condizioni sui cambiamenti di stato, se si usa una diversa filosofia di funzionamento di una rete sequenziale. Supponiamo che nella spira di reazione sia inserita una memoria a 1 bit e un interruttore normalmente aperto, il quale possa

essere chiuso per la durata di un opportuno segnale esterno, detto impulso di sincronismo, CP, **Fig. 3.10a**. Il principio di funzionamento di questa nuova architettura è il seguente:



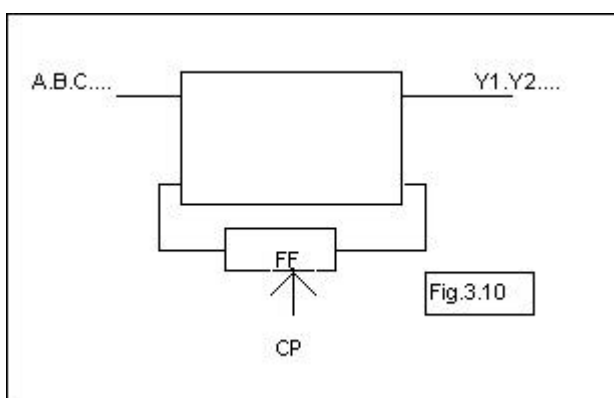
- 1) si effettua un cambiamento di stato di entrata, *senza alcun vincolo* sull'ordine in cui le variabili cambiano stato. Tale cambiamento provocherà il cambiamento di R e di Y. Si ignora Y;
- 2) si dà un CP, che deve durare solo il tempo necessario affinché il valore di

R si trasferisca nella memoria;

3) si legge il valore definitivo dell'uscita Y.

A questo punto, si può effettuare un altro passo nella sequenza di entrata.

Le reti sequenziali che così operano sono dette *sincrone*, poiché il loro funzionamento è scandito dagli impulsi CP, generati da un oscillatore a frequenza fissa (clock).



Si osservi che in una rete sincrona la spirale di reazione è *sempre aperta*, e la rete ha un comportamento sempre combinatorio, tranne in presenza di CP; le differenze nei tempi di propagazione dei segnali non avranno nessun effetto. Inoltre, non è ora necessario ricorrere all'artificio, usato nel caso asincrono, di

considerare il ritardo di propagazione concentrato nella spirale di reazione (artificio che serviva appunto ad operare su un circuito a spirale aperta). Per lo stesso motivo, R e r possono avere valore logico diverso, e non ha più senso distinguere fra stati instabili e stabili, poiché *tutti gli stati sono stabili*.

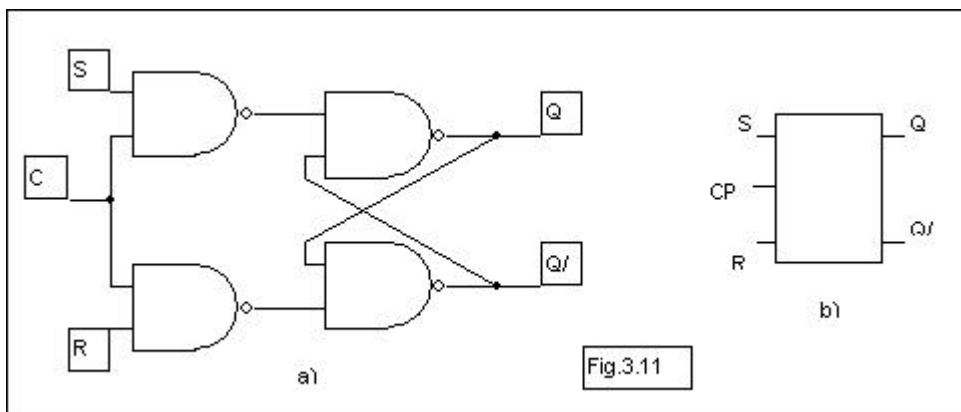
Il funzionamento sincrono è quello più diffuso, e di questo ci occuperemo nel seguito. In pratica, *l'interruttore* e la *memoria* sono realizzati mediante un FF "sincronizzato" che, sotto l'azione di CP, copia in uscita lo stato interno R, memorizzandolo. Lo schema a blocchi di una rete sequenziale sincrona è mostrato in **Fig. 3.10**. Naturalmente, possono esserci più spire di reazione, ciascuna col suo FF di memoria, tutti triggerati dallo stesso CP.

L'analisi di una rete sincrona consiste nella ricerca della sequenza degli stati di uscita per una assegnata sequenza di entrata. Più precisamente, noto lo stato delle entrate e lo stato di uscita dei FF, si vuole sapere in quale stato si porterà l'uscita dopo un impulso di sincronismo.

La sintesi di una rete sincrona consiste nel progettare la rete combinatoria che genera la sequenza di entrata ai FF necessaria per ottenere la richiesta sequenza di uscita.

Pertanto, il FF è l'elemento più importante di una rete sincrona. Poiché la sua conoscenza è preliminare sia al processo di analisi che a quello di sintesi, nei paragrafi seguenti illustreremo dapprima i vari tipi di FF sincronizzati. In questi,  $Q^n$  indica lo stato del FF *prima* dell'n-mo impulso di sincronismo (stato presente del FF);  $Q^{n+1}$  (stato seguente) indica lo stato del FF *dopo* l'n-mo CP.

### 3.5 IL FLIP-FLOP SR SINCRONIZZATO



Lo schema di un tale FF ed il suo simbolo circuitale sono in **Fig. 3.11**. Esso è costituito da un FF SR (noto anche come latch) e da due NAND di controllo. Questi due NAND fanno sì che le entrate S e R siano attive solo quando CP=1, allorquando tali NAND si comportano come due inverter per le rispettive entrate S e R. La tavola caratteristica di questo FF dovrebbe quindi essere identica a quella dell'SR non sincronizzato. In realtà, differisce per lo stato di entrata 11, che ora dà luogo ad uno stato seguente indefinito. Infatti, quando S=R=CP=1 entrambe le uscite sono a 1; quando CP torna a 0, i due ingressi al latch devono andare entrambi a 1 ma, a causa del diverso ritardo di propagazione dei due NAND, non vi arrivano contemporaneamente. Non essendo possibile prevedere quale dei due, di volta in volta, giungerà prima a 1, non si può predire lo stato seguente del FF, che potrà essere indifferentemente 1 o 0. Quindi, anche nell'SR sincronizzato lo stato di entrata 11 va evitato. Si osservi che durante l'intervallo di tempo in cui CP=1 le entrate al FF non devono cambiare: se così non è, il FF si fermerà nello stato di uscita relativo allo stato di entrata che è presente immediatamente prima che CP diventi 0.

S	R	Q <sub>n+1</sub>
0	0	Q <sub>n</sub>
0	1	0
1	0	1
1	1	?

a)

Q <sub>n</sub>	Q <sub>n+1</sub>	S	R
0	0	0	NI
0	1	1	0
1	0	0	1
1	1	NI	0

b)

**Fig.3.12**

La tavola caratteristica di questo FF è in **Fig. 3.12a**. In **Fig. 3.12b** è riportata invece la "tavola delle eccitazioni", la quale dà gli stati di entrata (eccitazioni) in corrispondenza di assegnate transizioni di uscita. Questa tavola è di lettura più immediata quando si effettua la sintesi di una rete sincrona. Peraltro, entrambe

le tavole danno la medesima informazione.

### 3.6 IL FLIP-FLOP JK

In questo FF viene eliminata l'indecisione nell'uscita che si ha nel FF SR quando  $S=R=1$ . Lo schema del JK ed il suo simbolo circuitale sono riportati in **Fig. 3.13**. Le tavole relative sono in **Fig. 3.14a e 3.14b**.

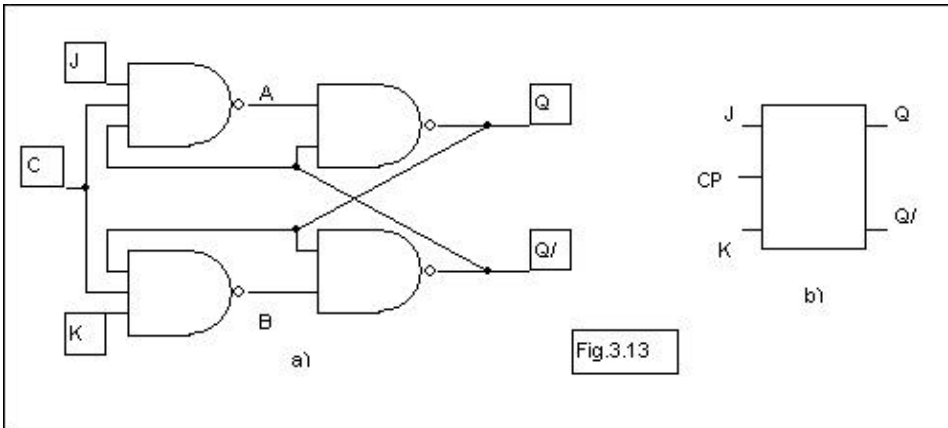


Fig.3.13

**3.14a e 3.14b.**

Esaminando la rete, ci si può convincere che quando  $J=K=1$  lo stato di uscita del FF viene ribaltato dopo che CP è

J	K	$Q_{n+1}$	$Q_n$	$Q_{n+1}$	J	K
0	0	$Q_n$	0	0	0	NI
0	1	0	0	1	1	NI
1	0	1	1	0	NI	1
1	1	$Q_n'$	1	1	NI	0

Fig.3.14

andato a 1. Infatti, supponiamo che, prima di CP, l'uscita sia 10. Poiché  $CP=0$ , le uscite A e B sono a 1. Quando  $CP \rightarrow 1$ , segue che  $B \rightarrow 0$  (dopo  $\delta t$  secondi) e quindi  $\bar{Q} \rightarrow 1$  (dopo  $2\delta t$  secondi). Allora, anche  $Q \rightarrow 0$  (dopo  $3\delta t$  secondi).

Tuttavia, se a questo punto è ancora  $CP=1$ , le

spire di reazione sono ancora chiuse e il nuovo stato del FF (01) provocherà un ulteriore ribaltamento dell'uscita. Questi ribaltamenti continuano fino a quando le spire vengono aperte, cioè quando CP va a 0; ma quando questo accade è impossibile prevedere lo stato finale di uscita. L'inconveniente potrebbe essere evitato controllando la durata del CP in modo tale da consentire un unico ribaltamento; tuttavia, data la brevità di tale tempo (dell'ordine del ritardo di propagazione delle porte), la condizione è troppo stringente.

Nel paragrafo seguente si descrive una versione modificata del JK che non richiede nessun vincolo sulla durata di CP.

Anche per il JK vale la considerazione fatta per l'SR, cioè lo stato di entrata non deve cambiare mentre  $CP=1$ .



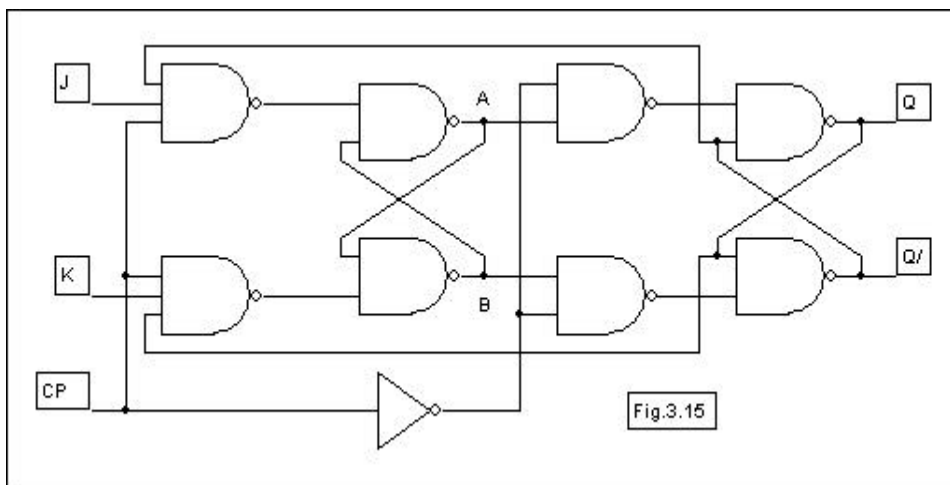
### 3.7 IL FLIP-FLOP JK MASTER-SLAVE

Esso è costituito da due FF in cascata, con reazione dal secondo al primo, come mostrato nello schema di **Fig. 3.15**. Il primo FF si chiama master, il secondo slave; tale nomenclatura è chiaramente suggerita dalla relazione funzionale fra i due FF, come ora vediamo.

Quando è  $CP=1$ , è facile verificare che il master si comporta come un JK, mentre lo slave è congelato nel proprio stato (a causa dell'inverter, per lo slave è  $CP=0$ ). Quindi, lo stato di uscita non può cambiare finché  $CP=1$ .

Quando CP diventa 0, lo stato del master viene congelato, mentre lo slave è abilitato e si comporta

come un SR le cui entrate sono le uscite A e B del master.



In conclusione, quando  $CP=1$  le entrate JK definiscono lo stato del master;

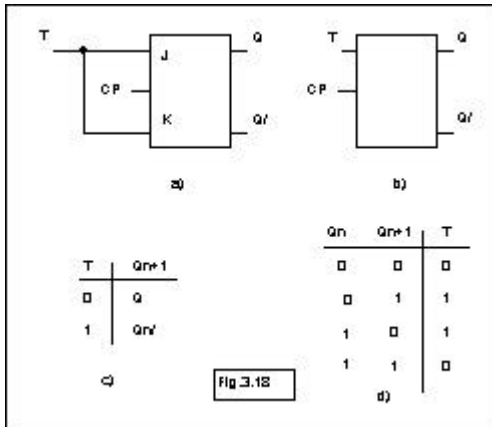
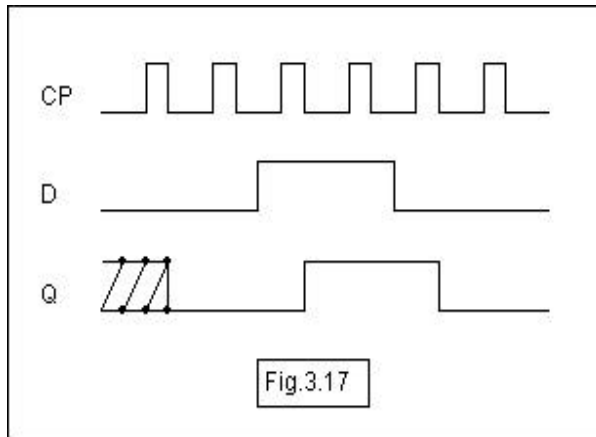
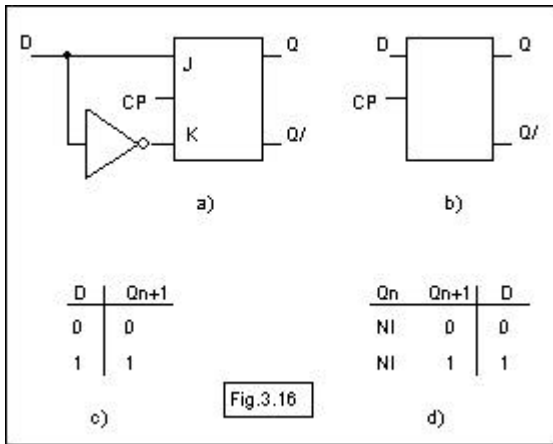
sul secondo fronte del CP, lo stato del master viene trasferito in uscita allo slave.

Si noti che le spire di reazione ingresso-uscita sono sempre aperte, pertanto non potrà instaurarsi la condizione di oscillazione, quando  $J=K=1$ , descritta nel paragrafo precedente.

Le tavole del JK MS ed il simbolo circuitale sono identici a quelli del JK semplice.

Ancora una volta, le entrate devono essere stabili finché  $CP=1$ .

### 3.8 IL FLIP-FLOP D E IL FLIP-FLOP T

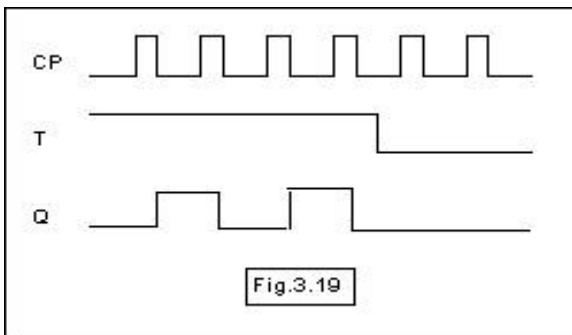


Il FF D è una variante del JK, come si vede nella **Fig. 3.16a**. La **Fig. 3.16b** mostra il simbolo circuitale. La tavola caratteristica è in **Fig. 3.16c**, quella delle eccitazioni in **Fig. 3.16d**.

Questo FF trasferisce in uscita lo stato di entrata dopo un CP. Si introduce, quindi, un ritardo ingresso-uscita (D=Delay) dipendente da CP, come mostrato in **Fig. 3.17**.

### 3.17.

Schema e simbolo circuitale del FF T sono in **Fig. 3.18a** e **3.18b**. Le tavole relative sono in **Fig. 3.18c,d**. Come illustrato dal **Fig. 3.19**, se T=1 ogni impulso di clock provoca un ribaltamento dello stato di uscita del FF (T=Toggle).



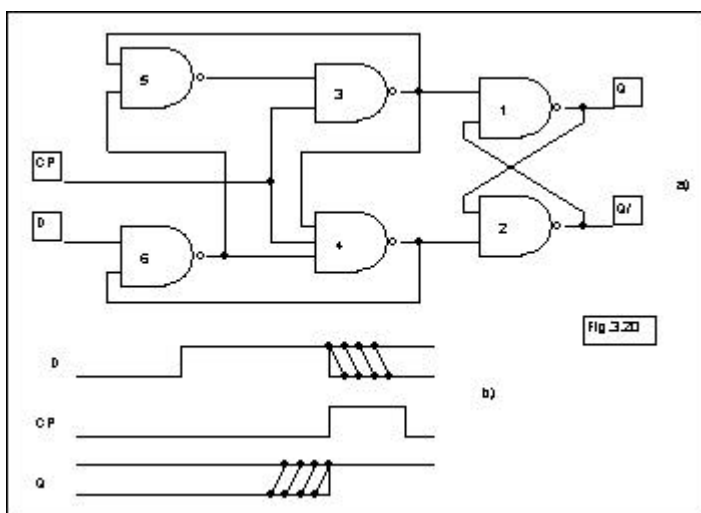
Questo tipo di FF non è disponibile commercialmente come tale, come è ovvio essendo direttamente ricavabile dal JK, che deve essere di tipo MS.

### 3.9 IL FLIP-FLOP EDGE-TRIGGERED

Come dice il nome, questo FF è sensibile ad un fronte del CP (quello  $\downarrow$  o quello  $\uparrow$ ), non alla sua durata; quando questo fronte si presenta, il FF commuta e una logica interna disabilita (dopo pochi nsec) le porte di entrata al FF stesso che, pertanto, resta isolato dalle entrate fino all'arrivo di un'altra transizione attiva del clock.

In questo modo, si elimina il rischio che un imprevisto cambiamento delle entrate (per esempio a causa di disturbi) possa dar luogo ad uno stato di uscita errato.

In Fig. 3.20a è riportato lo schema di un FF D edge-triggered. Il suo funzionamento viene ora descritto.



Sia  $D=1$ . Se  $CP=0$ , le uscite delle porte 3 e 4 sono a 1 e l'uscita del FF è in uno stato ignoto. L'uscita della porta 6 è a 0, quella della porta 5 è a 1.

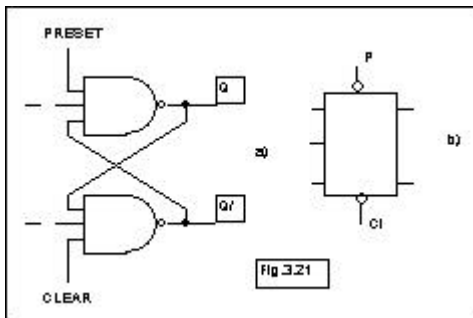
Quando  $CP \rightarrow 1$ , l'uscita della porta 3 va a 0 e Q va a 1; lo 0 all'uscita della porta 3 inibisce le porte 4 e 5, per cui lo stato di D, anche se cambia, non può più influenzare il latch.

Sia ora  $D=0$ . Quando  $CP=0$ , le porte 3, 4 e 6 hanno l'uscita a 1. Quando  $CP \rightarrow 1$ , l'uscita della porta 4 va a 0, congelando lo 0 nel latch e inibendo la porta 6 contro eventuali variazioni di D.

### 3.10 INGRESSI DI PRESET E CLEAR AL FLIP-FLOP

Occasionalmente, durante il funzionamento di una rete sincrona nasce la necessità di forzare i flip-flop in uno stato prestabilito. Questo accade normalmente all'accensione della rete, quando i FF si portano in uno stato ignoto.

A questo scopo, i FF hanno due ingressi accessori al latch (asincroni), chiamati preset e clear (o reset), **Fig. 3.21**. Talvolta, è presente il solo clear.



Ponendo temporaneamente il preset a 0, il FF viene forzato nello stato 1.

Ponendo, invece, temporaneamente il clear a 0, il FF viene forzato nello stato 0.

Nella stessa figura è indicato il simbolo.

### 3.11 ANALISI DI UNA RETE SEQUENZIALE SINCRONA

L'analisi di una rete sincrona ha lo scopo di determinare lo stato di uscita della rete dopo un impulso di clock, CP, una volta noti lo stato di entrata e lo stato presente dei FF, (cioè lo stato dei FF prima dell'applicazione dello stesso CP), **Fig. 3.10**. Poiché prima e dopo CP la spirale di reazione è aperta, dalla **Fig. 3.10** si vede che si tratta di analizzare:

1. una rete combinatoria, le cui entrate sono i segnali esterni A, B, .. e lo stato interno presente  $Q^n_i$  dei flip-flop, e le cui uscite sono le eccitazioni  $J_i, K_i$  dei FF (cioè le entrate a questi ultimi prima del CP) e le uscite  $Y_1, Y_2...$  della rete sincrona;
2. uno o più FF di memoria, di cui è nota la tavola caratteristica.

Pertanto, il procedimento di analisi si sviluppa nei seguenti passi:

- dalla rete combinatoria si ricavano le equazioni di eccitazione dei FF

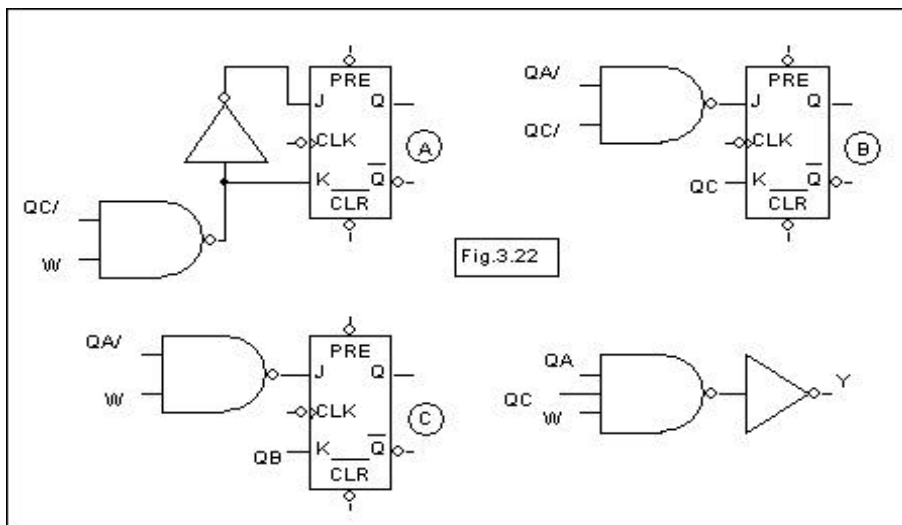
$$J_i, K_i = f(A, B, \dots, Q^n_1, Q^n_2, \dots)$$

le quali forniscono le entrate ai FF prima di CP;

- note le eccitazioni, dalla tavola caratteristica dei FF si ricava lo stato seguente dei FF,  $Q^{n+1}_i$ ;
- noti gli stati seguenti, dalle equazioni delle uscite

$$Y_1, Y_2, \dots = f(A, B, \dots, Q^{n+1}_1, Q^{n+1}_2, \dots)$$

si calcola lo stato di uscita della rete.

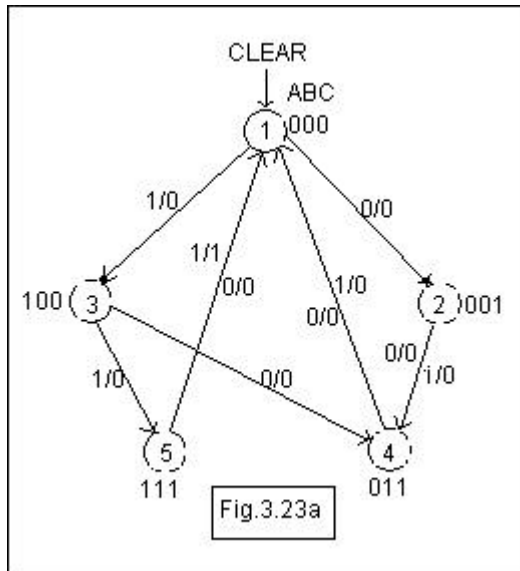


Il risultato dell'analisi si riassume nel "diagramma degli stati", nel quale vengono numerati gli stati interni della rete e vengono indicate le possibili transizioni.

Poiché ogni FF ha due

stati, nel diagramma degli stati compariranno  $2^n$  stati, essendo n il numero dei FF.

Poiché le variabili di entrata sono indipendenti dallo stato dei FF, le transizioni che si dipartono da ogni stato saranno  $2^m$ , essendo m il numero di tali variabili: peraltro alcune transizioni possono coincidere.



Illustriamo con un esempio quanto ora detto.

Consideriamo la rete di **Fig. 3.22**, avente un ingresso W, una uscita Y, e tre FF JK MS di memoria. La rete è già disegnata in modo opportuno per l'analisi.

Le equazioni di eccitazione (che danno le entrate ai FF prima che ad essi venga applicato l'n-mo impulso di clock) sono le seguenti

$$J_A = \bar{Q}^n_C W \qquad J_B = Q^n_C + Q^n_A$$

$$J_C = Q^n_A + \bar{W}$$

$$K_A = Q^n_C + \bar{W} \qquad K_B = Q^n_C \qquad K_C = Q^n_B$$

L'equazione di uscita (che dà lo stato dell'uscita dopo l'applicazione dell'n-mo impulso di clock) è

$$Y = Q^{n+1}_A \cdot Q^{n+1}_C \cdot W$$

Iniziamo ora l'analisi, partendo dallo stato interno presente 000, raggiunto per esempio mediante un Clear iniziale (questo è quindi lo stato presente al momento di inviare il primo CP). Tale stato è numerato 1 nel diagramma di **Fig. 3.23a**.

Se  $W = 0$ , dalle equazioni di eccitazione si ha

$$\begin{aligned} J_A &= 0 & J_B &= 0 & J_C &= 1 \\ K_A &= 1 & K_B &= 0 & K_C &= 0 \end{aligned}$$

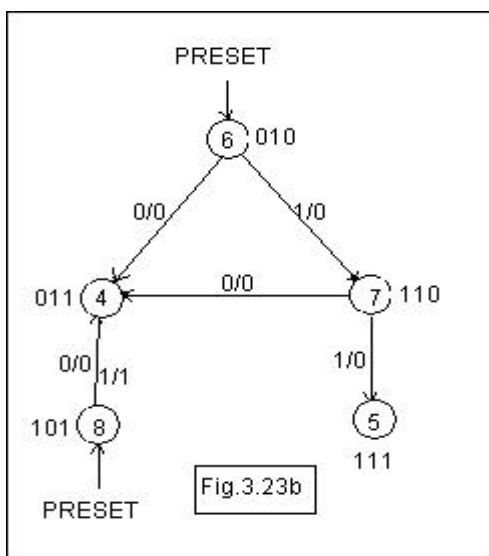
Con queste eccitazioni, dalla tavola caratteristica del JK si ricavano gli stati seguenti (all'impulso di clock)

$$Q^{n+1}_A = 0 \qquad Q^{n+1}_B = 0 \qquad Q^{n+1}_C = 1$$

Una volta noti gli stati seguenti, dall'equazione di uscita si ha  $Y = 0$ .

La rete ha raggiunto lo stato 2 del diagramma. La transizione esaminata è rappresentata dalla freccia che unisce i due stati. La notazione 0/0 a fianco della freccia indica rispettivamente lo stato di entrata prima di CP e lo stato di uscita dopo CP.

Partendo ancora dallo stato 1, si considera ora la seconda transizione possibile, quella con  $W = 1$ . Si trova facilmente che nello stato 3 raggiunto da tale transizione, lo stato seguente è 100 e lo stato di uscita è 0.



Procedendo come sopra, si trova che partendo da 2 si finisce comunque in 4; partendo da 3 si finisce o in 4 o in 5; partendo infine da 4 o da 5 si finisce comunque in 1.

A questo punto, la sequenza delle transizioni è chiusa, e si vede che restano esclusi gli stati interni 010, 110, 101.

Per completare l'analisi, occorre reinizializzare la rete portandola in uno di questi stati, il che può essere realizzato mediante l'ingresso di Preset di ciascun FF.

Inizializziamo la rete nello stato 010, chiamandolo 6.

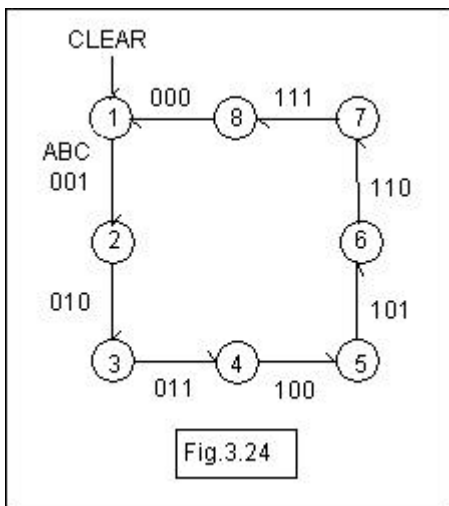
Procedendo come sopra, l'analisi viene rapidamente completata; **Fig. 3.23b**.

Si noti in questo esempio che se il funzionamento della rete inizia con un reset, gli stati 6, 7, 8 non verranno mai percorsi.

### 3.12 SINTESI DI UNA RETE SINCRONA

Essendo la sintesi il processo inverso dell'analisi, per effettuarla si partirà, evidentemente, da un diagramma degli stati per arrivare alla rete combinatoria che fornirà la sequenza delle eccitazioni dei flip-flop. Il procedimento viene illustrato in dettaglio con un semplice esempio. Si vuole sintetizzare un contatore sincrono binario a tre bit; come verrà meglio illustrato nel seguito, il contatore è una rete il cui stato di uscita dopo  $n$  impulsi di clock è il numero  $n$  espresso in codice binario naturale; nel nostro caso, partendo dallo stato 000 e inviando 7 impulsi di clock le uscite saranno, dopo ogni impulso

001, 010, ..... 111



L'ottavo impulso di clock deve rimettere le uscite nello stato 000 (questo perché l'8 in binario si scrive 1000, che eccede la capacità del contatore a tre bit (overflow), ed i tre bit meno significativi sono appunto 000).

La richiesta del problema è rappresentata nel diagramma degli stati di Fig. 3.24. Si suppone di entrare nella sequenza dallo stato 1, con un clear: lo stato interno della rete è in corrispondenza 000. Accanto ad ogni transizione uscente da ciascuno stato è indicato lo stato di uscita della rete dopo

CP (l'ingresso al contatore è il solo CP).

Il numero di stati interni della rete è 8; il numero di FF necessari è quindi 3.

Occorre, ora, stabilire quale stato assegnare ai tre FF in corrispondenza dei rimanenti 7 stati interni

$Q_n$	$Q_{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

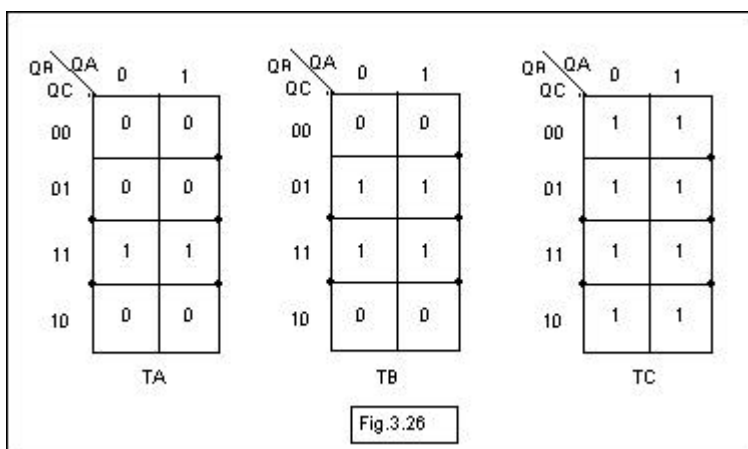
Fig. 3.25

della rete, e che tipo di FF usare. In linea di principio sia la scelta che l'assegnazione sono arbitrarie, ma è intuitivo che si otterranno soluzioni circuitali diverse sia per complessità che per prestazioni. Purtroppo, non sono note regole precise per individuare il tipo di FF e l'assegnazione che daranno luogo alla soluzione più semplice: ci si può solo affidare all'esperienza o a qualche tentativo.

L'assegnazione più conveniente degli stati interni dei FF è nel nostro caso

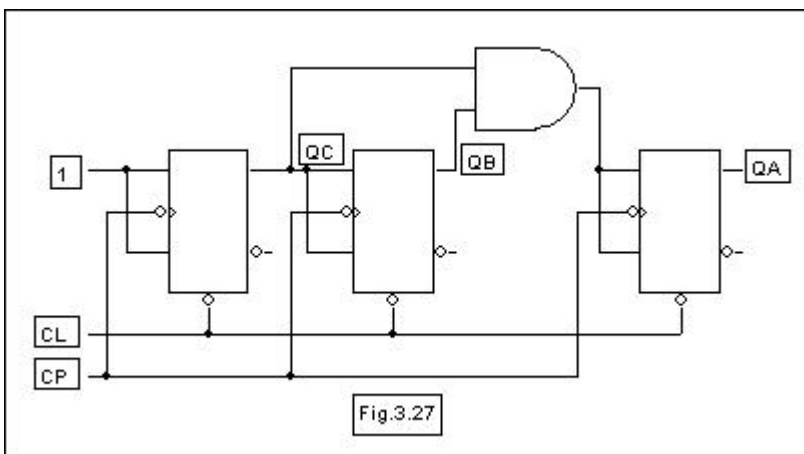


evidente: conviene scegliere gli stati seguenti coincidenti con le uscite, cosicché non sarà necessario hardware aggiuntivo per la decodifica (le uscite dei FF saranno anche le uscite del contatore).



Come FF si sceglie il tipo T, la cui tavola delle eccitazioni è in **Fig. 3.25**. Per sintetizzare le equazioni di eccitazione dei singoli FF si compilano delle mappe di Karnaugh, Fig. **3.26**: si noti che gli indici di riga e di colonna sono gli stati presenti dei FF. In ogni casella di tali mappe

compare il valore dell'eccitazione T che porta il FF nello stato seguente indicato nel diagramma degli stati. Il procedimento di compilazione delle mappe è il seguente:



compilazione delle mappe è il seguente:

- dal diagramma degli stati appare che, partendo dallo stato 1 (stato interno presente  $Q_A=0, Q_B=0, Q_C=0$ ), dopo un impulso di clock si transita nello stato seguente 001; cioè:

- 1- il FF A deve restare nello stato 0, e perché questo accada deve essere  $T_A=0$ , come appare dalla tavola delle eccitazioni. Questo valore di T è inserito nella casella 000 della mappa del FF A.
- 2- Idem per il FF B.
- 3- Il FF C deve invece transitare da 0 a 1: perché questo accada, deve essere  $T_C=1$ ; e questo valore viene posto nella casella 000 di C. Ecc.

- Dalle mappe si sintetizzano le equazioni

$$T_C = 1 \qquad T_B = Q_C \qquad T_A = Q_C Q_B$$

-3.26-

La rete è disegnata in **Fig. 3.27**.

Questo metodo può essere esteso per sintetizzare un contatore a n bit; si avrà in generale

$$T_i = Q_{i-1} Q_{i-2} \dots Q_0 \quad i = 0, 1, 2, \dots, n-1$$

o anche

$$T_i = Q_{i-1} T_{i-1}$$

Se ne riparla più avanti.

Si suggerisce come esercizio la sintesi di un contatore a 3 bit up-down.

### 3.13 I CONTATORI ED I REGISTRI

Vi sono due esigenze che sono comuni ad una vastissima gamma di applicazioni dell'elettronica digitale: la registrazione, temporanea o permanente, di informazione sotto forma di parola binaria, ed il conteggio di impulsi, il cui numero totale o la cui frequenza rappresenta l'informazione che interessa.

Circuiti integrati che soddisfano queste esigenze sono disponibili commercialmente e possono essere adoperati come blocchi funzionali per costruire sistemi più complessi, senza bisogno di sintetizzare la rete nel suo insieme.

Si è già visto che una cella elementare di memoria, nella quale è possibile registrare in modo permanente un bit di informazione, è il flip-flop. Consideriamo dapprima il caso che si voglia registrare una sola parola di  $k$  bit. Un "registro" per tale parola sarà un array unidimensionale di  $k$  flip-flop. L'accesso ai bit del registro può avvenire in due modi:

a) i  $k$  bit della parola vengono scritti serialmente nei FF del registro: precisamente, i bit vengono fatti entrare uno alla volta nel primo FF e fatti scorrere lungo la catena fino a quando ogni bit ha raggiunto il proprio posto. Si ha così un "registro a scorrimento" (shift register). Naturalmente, nello shift register la lettura o la scrittura dell' $i$ -mo bit della parola può avvenire solo passando attraverso i bit che lo separano dall'entrata o dall'uscita del registro stesso.

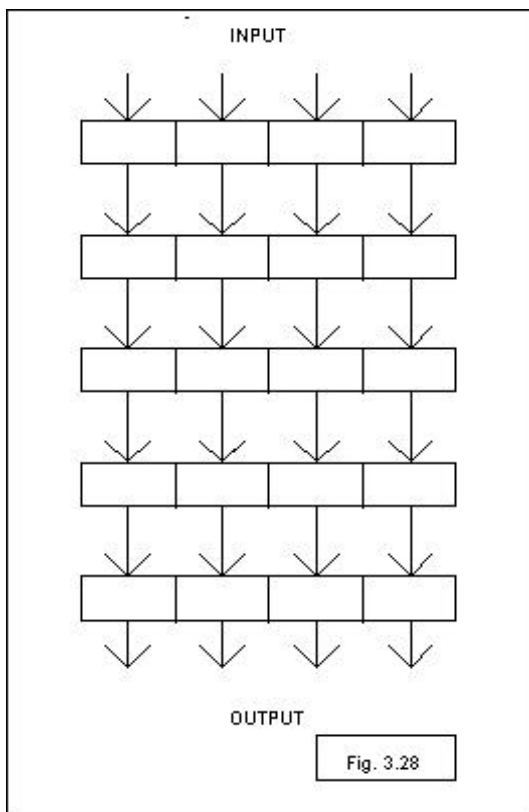
b) I  $k$  bit della parola vengono scritti in parallelo nei  $k$  FF del registro.

Consideriamo ora il caso generale della registrazione di  $N$  parole di  $k$  bit. Una "pagina di memoria" di  $N$  parole sarà un insieme di  $N$  registri, organizzata in modo tale da poter accedere ai singoli registri.

Se gli  $N$  registri sono assemblati in modo tale che ognuno sia accessibile indipendentemente dagli altri, si ha una memoria ad accesso casuale (Random Access Memory, RAM). Essa è descritta in dettaglio più avanti.

Se invece i registri sono accessibili in sequenza, si ha una memoria ad accesso sequenziale. In questo caso si parla di memorie FIFO (First In First Out) quando la prima parola scritta è anche la prima che può essere letta. Lo schema di principio di una FIFO di 5 parole di 4 bit è mostrato in

**Fig. 3.28.**



Si parla invece di memorie LIFO (Last In First Out), quando l'ultima parola scritta è anche la prima a poter essere letta. Lo schema di principio è mostrato in Fig. 3.29.

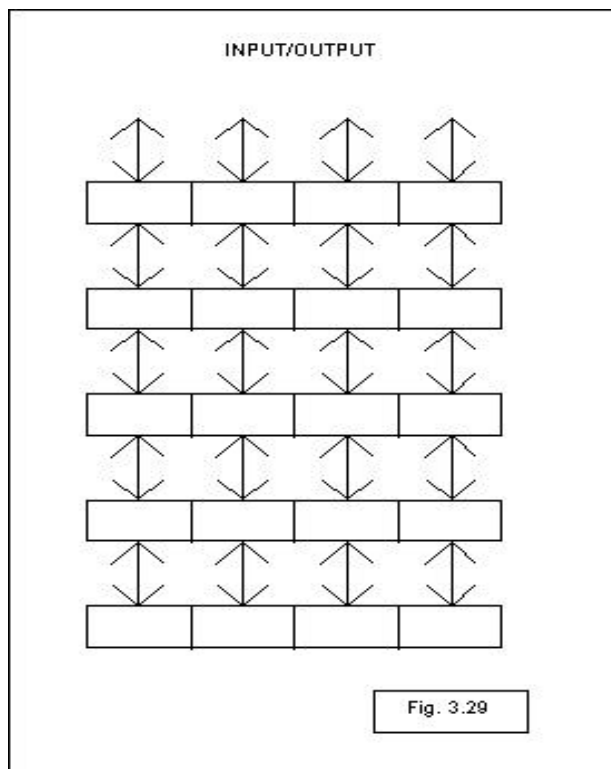
Un contatore è anch'esso costruito mediante FF, perché è in definitiva un registro nel quale viene immagazzinata temporaneamente una informazione (il numero di impulsi entrati fino all'istante  $t_0$ ), aggiornabile man mano che altri dati (gli impulsi che arrivano dopo  $t_0$ ) si presentano all'ingresso.

Il contatore elementare è quello che conta fino a 1. Esso è costituito da una rete avente una uscita che è inizializzata a 0 e diventa 1 dopo il primo CP da

contare. Il secondo CP eccede la capacità di conteggio del contatore (overflow), e riporta l'uscita a 0.

Contatori con maggiore capacità di conteggio possono evolvere da questo contatore elementare secondo due distinti principi:

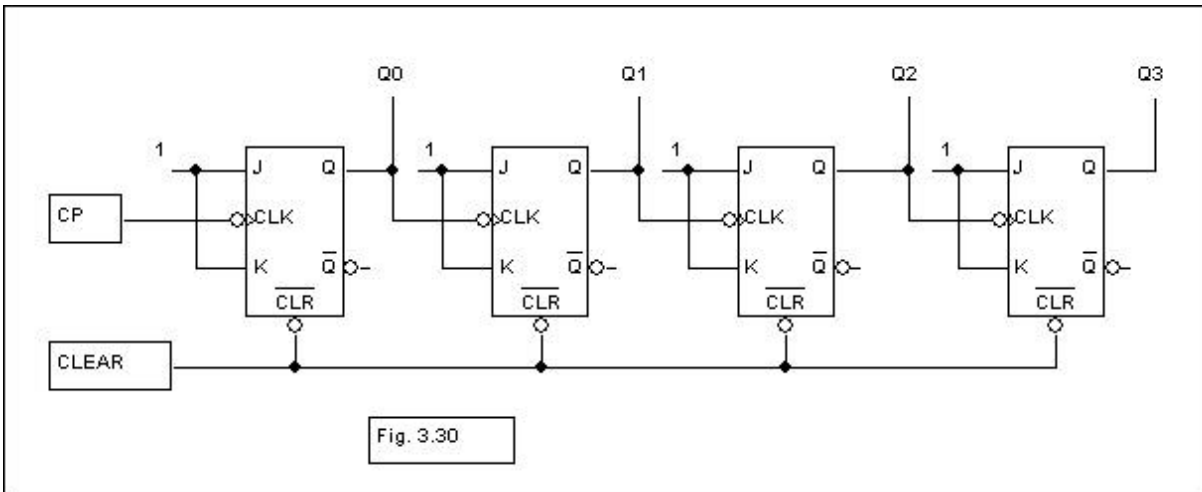
- conteggio asincrono: si realizza assemblando in cascata  $n$  contatori elementari, cosicché ciascuno di essi conta gli impulsi di uscita del precedente.
- Conteggio sincrono: si realizza assemblando  $n$  FF con ingresso del CP in parallelo, ma con una opportuna rete combinatoria che programma le transizioni.



Nei paragrafi seguenti si esaminano entrambe le soluzioni.

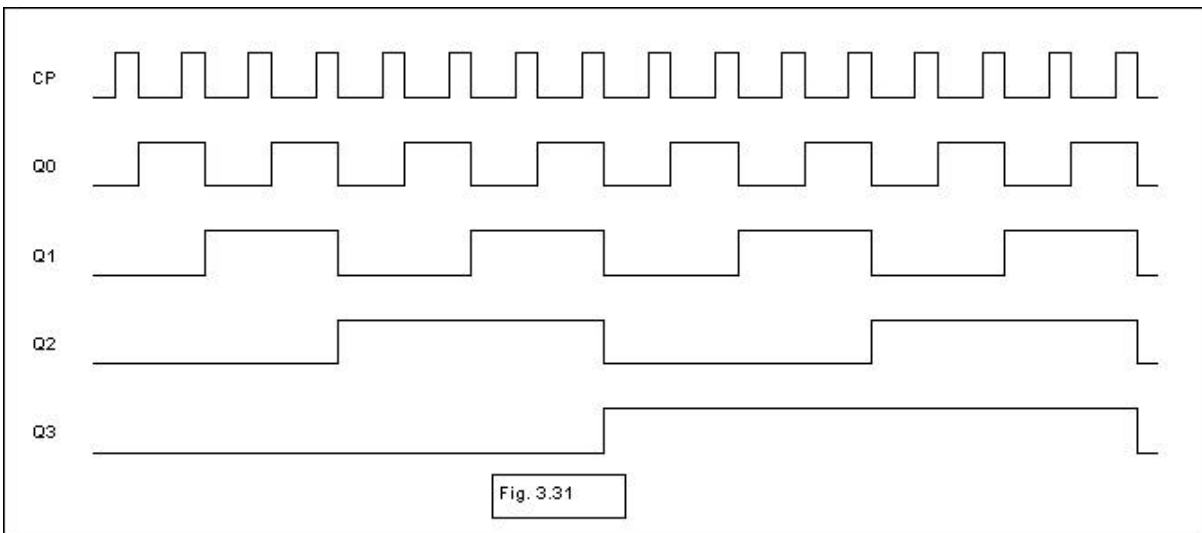
I contatori commercialmente disponibili su singolo integrato sono costituiti generalmente da 4 FF; a questa struttura ci riferiremo nel seguito.

### 3.14 IL CONTATORE ASINCRONO



Lo schema del contatore (ripple counter) è in **Fig. 3.30**: i flip-flop sono TMS, con  $T=1$ . Per inciso, questo schema è un risultato euristico, nel senso che non esiste una procedura per sintetizzarlo.

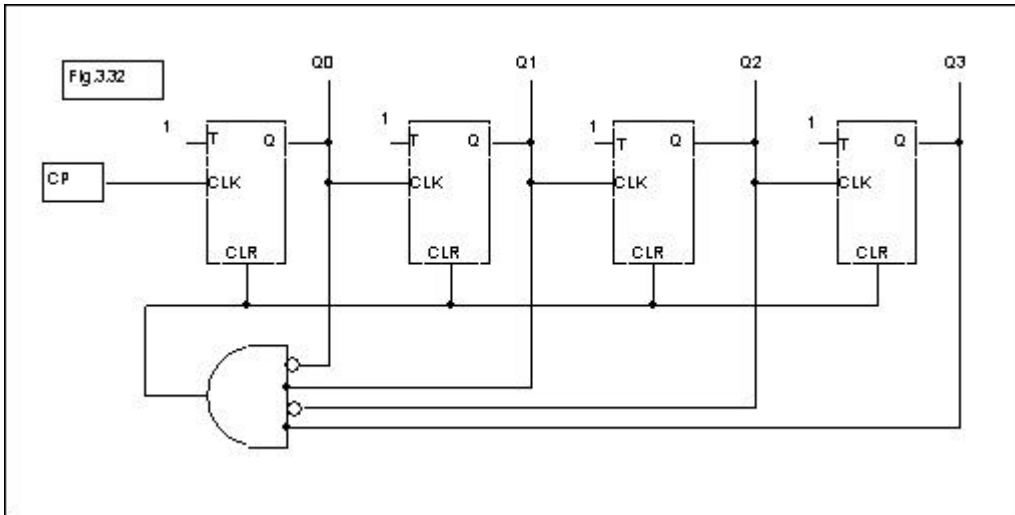
In **Fig. 3.31** sono disegnati gli stati dei FF durante un treno di impulsi di ingresso, dopo un reset iniziale. Si può agevolmente verificare che lo stato di uscita del contatore dopo  $k$  impulsi di ingresso è il numero binario naturale  $k$ .



Un contatore binario a 4 stadi può contare fino a  $2^4 - 1 = 15$  impulsi (1111, in binario naturale); il successivo impulso, corrispondente a 10000 binario, richiederebbe un quinto FF e pertanto eccede la capacità di conteggio del contatore a 4 bit, il quale riporta lo stato dei quattro bit meno significativi, cioè 0000.

Si dice che un contatore binario naturale a  $n$  stadi è un contatore in modulo  $2^n$ . È possibile modificare un contatore asincrono per contare in modulo  $N < 2^n$ . Per ottenere questo, basta far sì che la scala

di conteggio venga resettata quando arriva l'N-mo impulso. Pertanto, basta decodificare il numero N in uscita e usare l'uscita dal decoder come reset.



In Fig. 3.32 è mostrato un contatore decadico.

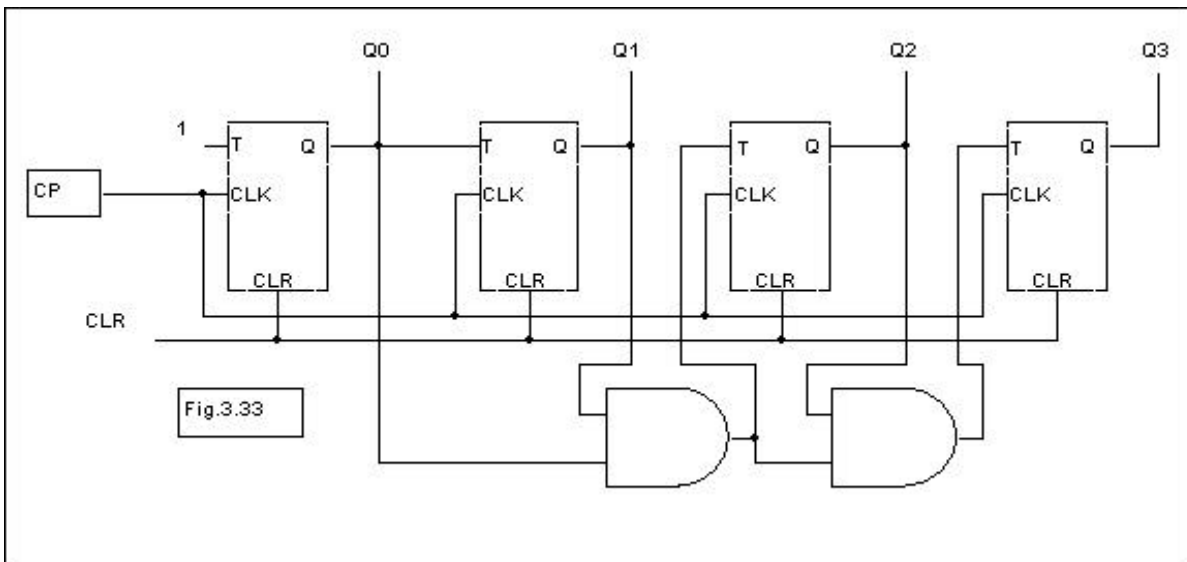
### 3.15 IL CONTATORE SINCRONO

Il contatore asincrono è strutturalmente semplice, ma ha generalmente bassa velocità di conteggio: infatti, per evitare equivoci, la lettura del contatore deve essere fatta dopo che tutti i flip-flop della catena hanno raggiunto lo stato seguente all'impulso n-mo di sincronismo. Solo allora si può inviare l'impulso (n+1)-mo. Ora, dalla Fig. 3.30 è evidente che il FF i-mo cambia stato dopo che hanno cambiato stato, uno dopo l'altro, tutti i FF precedenti. Il caso più sfavorevole si verifica quando tutti i FF della catena devono cambiare stato. Pertanto, il periodo di clock non può essere inferiore a

$$T_D = nt_{pd}$$

essendo n il numero di FF e  $t_{pd}$  il ritardo di propagazione di un singolo FF. La massima frequenza di conteggio sarà quindi  $f_{max}=1/T_D$ .

Prestazioni migliori si ottengono con il conteggio sincrono: gli impulsi da contare sono inviati contemporaneamente a tutti i FF della catena, e una opportuna rete di controllo programma l'eccitazione dei singoli FF, cosicché lo stato di uscita sia quello corretto. Come si vedrà fra poco, il risultato è un ritardo di propagazione inferiore a quello della catena asincrona, o addirittura indipen-



dente dalla lunghezza della catena stessa.

La sintesi di un contatore sincrono a 3 bit è già stata effettuata nel paragrafo 3.13, e può ovviamente essere generalizzata al caso di n FF; l'equazione di eccitazione del FF i-mo può infatti essere scritta, come già visto



$$T_i = Q_{i-1}Q_{i-2} \dots Q_0$$

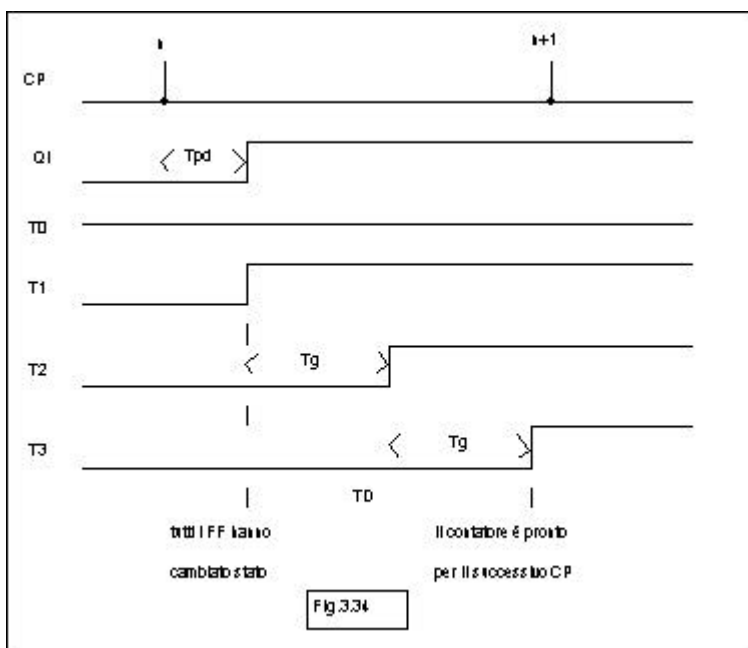
o anche

$$T_i = Q_{i-1}T_{i-1}$$

Utilizzando questa seconda espressione, lo schema di un contatore a 4 bit è mostrato in **Fig. 3.33**.

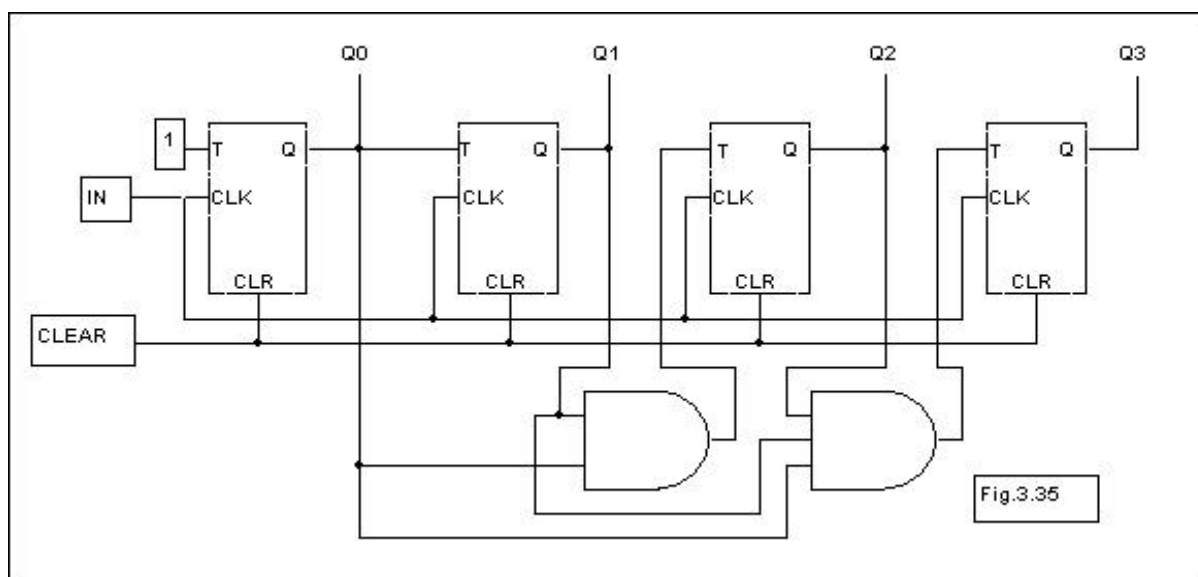
In questo schema, conosciuto come "serial carry", il massimo ritardo di propagazione è

$$T_D = t_{pd} + (n-2)t_g$$



essendo n il numero di FF e  $t_g$  il ritardo di propagazione di una gate. Ciò si deduce dalla **Fig. 3.34**, che riporta le eccitazioni per i singoli FF nell'intervallo fra due CP, per stabilire  $T_D$ . Confrontando circuiti reali, si trova che il ritardo nel contatore sincrono serial-carry è circa la metà di quello asincrono.

Una versione più veloce di contato-



re (tipo "parallel carry") si implementa utilizzando la prima espressione di  $T_i$ , da cui deriva lo schema di **Fig. 3.35**. A differenza del serial carry, la transizione di un FF viene ora riportata ai successivi

-3.34-

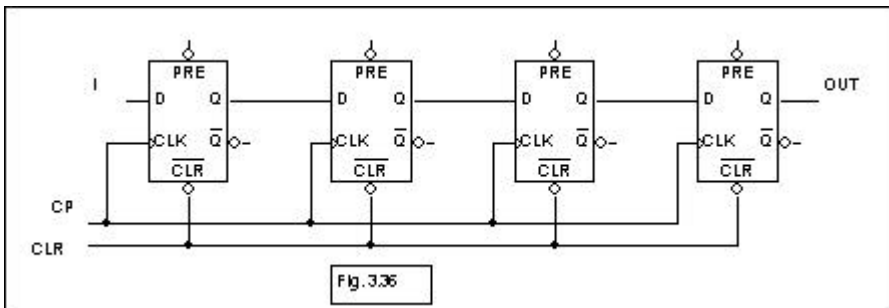
nella catena attraversando una sola gate. In altri termini, l'ingresso T di ciascun FF viene condizionato dalle uscite di tutti i FF precedenti in parallelo. Si può verificare che il ritardo massimo è

$$T_D = t_{pd} + t_g$$

minore del caso precedente, anzi indipendente dal numero dei FF.

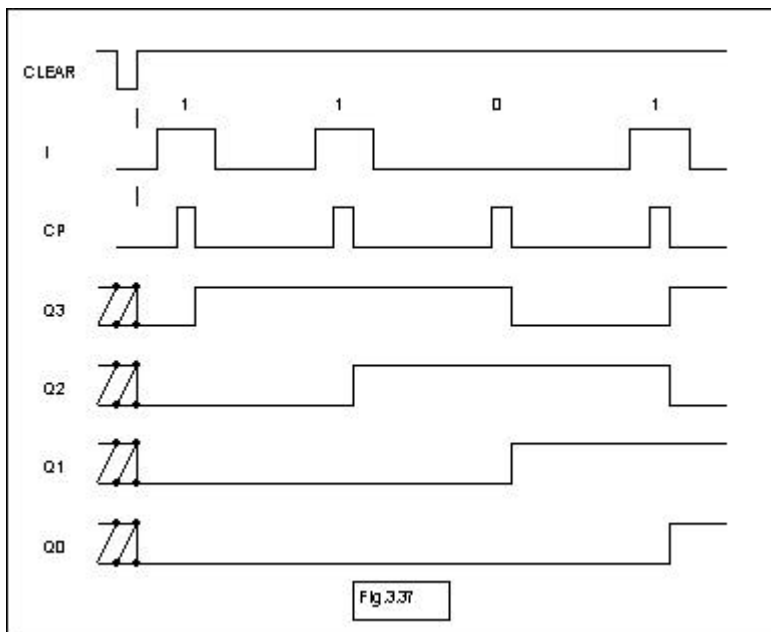
### 3.16 LO SHIFT REGISTER

Si è già detto che esso è un registro a n bit con una entrata ed una uscita, nel quale la registrazione di



una parola, ovviamente di n bit, avviene presentando i bit uno alla volta all'entrata, cominciando col bit meno significativo, facendoli quindi scorrere (me-

diante impulsi di clock) lungo i successivi flip-flop fino a riempire il registro.



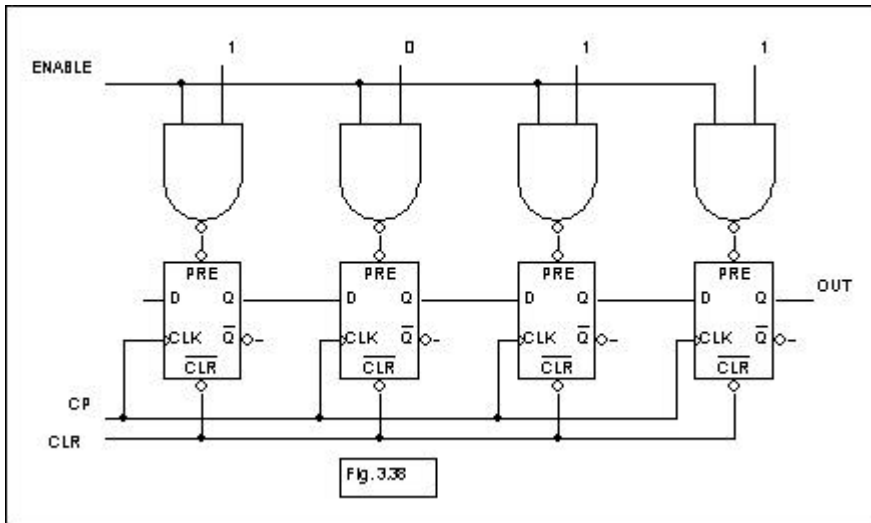
Lo schema dello shift register è in **Fig. 3.36**.

Per comprendere meglio il funzionamento del registro, supponiamo che dopo un reset iniziale venga presentata all'ingresso la parola 1011, in sincronismo con CP, come mostrato in **Fig. 3.37**, iniziando col bit meno significativo. Come mostrato nella stessa figura, ogni impulso di clock

spinge in un FF il bit presente all'ingresso del medesimo (che è quello di uscita del FF precedente); dopo 4 CP la parola è stata registrata.

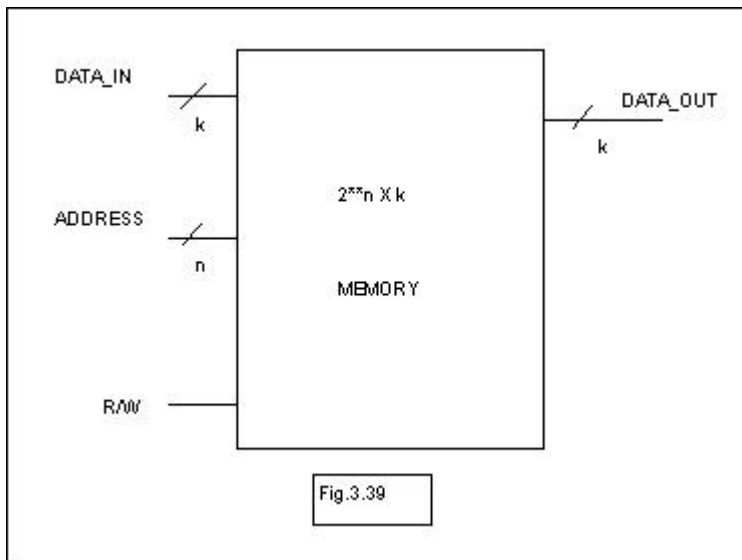
La lettura del contenuto dello shift register avviene inviando altri 4 impulsi di clock i quali forzeranno l'uscita da Q<sub>0</sub> dei 4 bit registrati.

Uno shift register può essere usato come un registro con scrittura in parallelo della parola, come richiesto per es. nelle FIFO e nelle LIFO: per questo si accede ai singoli FF dall'ingresso di preset. Lo schema è mostrato in **Fig. 3.38**. Dopo un reset iniziale, si dà un impulso positivo sull'ingresso "enable", il quale forza i singoli FF nello stato del bit presente al rispettivo ingresso di preset.



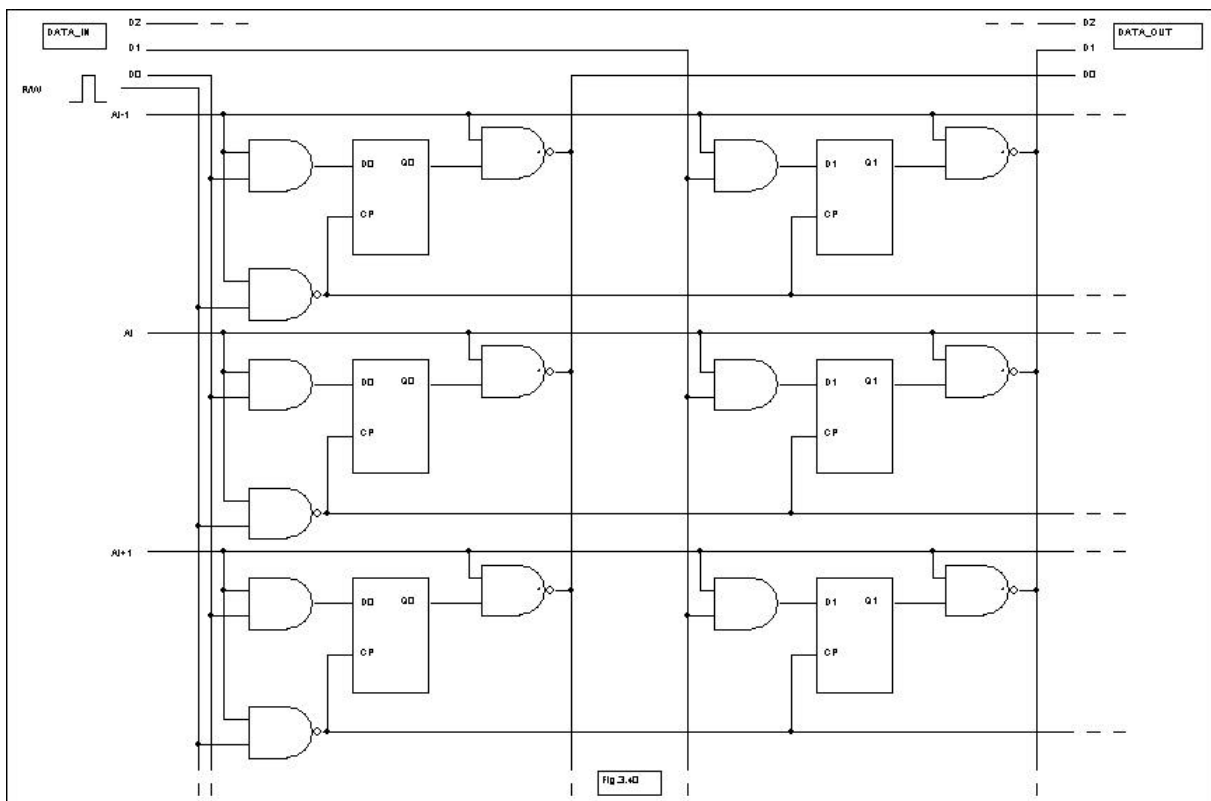
Talvolta, lo shift register può anche permettere la lettura parallelo, oltre che seriale, della parola, rendendo accessibili le 4 uscite  $Q_3$ ,  $Q_2$ ,  $Q_1$ ,  $Q_0$ , generalmente tramite gates abilitate da un segnale di strobe.

### 3.17 LA MEMORIA AD ACCESSO CASUALE



Una RAM (Random Access Memory) è un insieme di registri, o "locazioni di memoria", in ciascuno dei quali è possibile scrivere una parola binaria. La RAM è organizzata in modo tale che le locazioni siano accessibili direttamente e indipendentemente dalle altre, da cui il nome di memoria ad accesso casuale.

La struttura generale a blocchi di una pagina di memoria di  $2^n$  parole ciascuna di  $k$  bit è mostrata in **Fig. 3.39**.



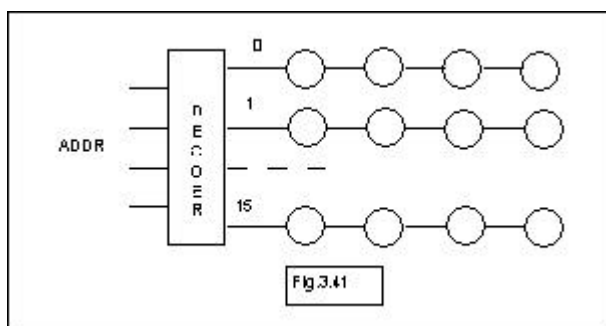
Per accedere ad una locazione occorre "indirizzarla", cioè occorre presentare sulle linee di indirizzo la parola in codice, di  $n$  bit, che individua la locazione stessa (l'indirizzo rappresenta il numero d'ordine della parola nella pagina di memoria). Questo indirizzo viene interpretato da un decoder

interno, che abilita l'accesso alla locazione indirizzata.

L'accesso ai k FF (che sono sempre di tipo edge-triggered) della locazione indirizzata avviene tramite k linee dati, talvolta distinte per la scrittura e la lettura, ma di norma uniche poiché le due fasi di scrittura e lettura non sono generalmente contemporanee.

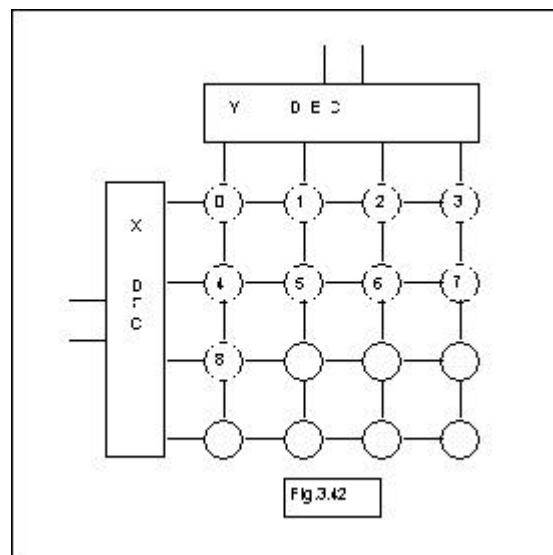
La decisione se scrivere o leggere è stabilita dallo stato della linea R/W (Read/Write).

La struttura interna della pagina di memoria è dettagliata nella Fig. 3.40, ove per semplicità le linee di ingresso e uscita sono separate. Il segnale  $A_j$  è il risultato della decodifica dell'indirizzo; cioè



quando si indirizza la j-ma locazione il segnale  $A_j$  va a 1 (mentre gli altri  $A_i, i \neq j$ , sono a 0): in tal modo, tutti i FF della locazione j-ma sono accessibili per la scrittura (il dato di ingresso viene scritto dall'impulso sulla linea R/W solo nella locazione j-ma), e per la lettura (solo le

uscite dei FF della locazione j-ma possono accedere al bus di uscita). Si noti che sulle linee di uscita è realizzato un OR cablato.



Per organizzazione della memoria si intende il numero di parole che possono essere contenute nella memoria stessa ed il numero di bit di ciascuna parola. Così, una memoria 64x8 contiene 64 parole, ciascuna di 8 bit. Sono disponibili (2002) RAM su singolo integrato fino a  $64 \cdot 10^6 \times 1$ , ma l'aumento della capacità è praticamente continua.

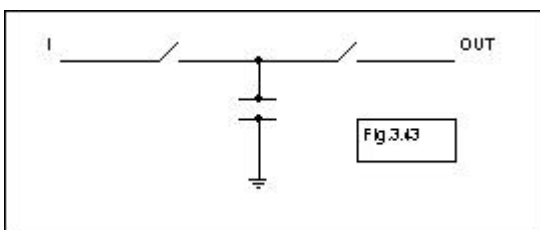
Nelle RAM l'indirizzamento delle singole parole può avvenire in due modi. Considerando, per esempio, una RAM 16x4, l'indirizzamento più ovvio è tramite

un decoder 4-a-16, come mostrato in Fig. 3.41. Tale decoder è integrato sullo stesso chip della RAM.

Questo tipo di indirizzamento diventa impraticabile se il numero di parole è grande. Si ricorre allora

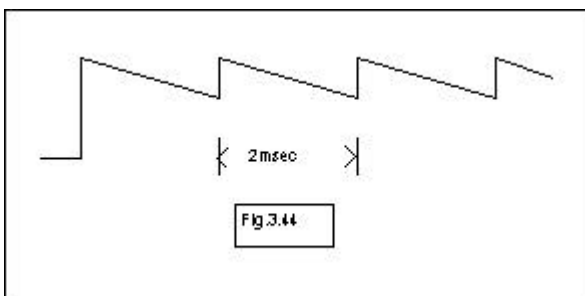
ad un indirizzamento per righe e colonne, **Fig. 3.42**. La memoria può essere ora pensata come una struttura tridimensionale, organizzata a piani di bit; cioè i bit dello stesso ordine in ogni parola sono su un piano e si hanno tanti piani (lungo l'asse z, normale al piano di figura) quanti sono i bit di una parola. Così, in una memoria 16x4 ci saranno 4 piani con 16 bit per piano.

La parola indirizzata è quella che si trova all'incrocio delle linee X,Y selezionate dai due decoder. Il vantaggio appare evidente: anziché un decoder con 16 uscite occorrono due decoder ciascuno avente  $\sqrt{16}$  linee di uscita.



Le RAM a stato solido hanno lettura non distruttiva (cioè, l'operazione di lettura non distrugge il contenuto della memoria). Il loro contenuto è però di tipo volatile, cioè esso viene perso se viene tolta l'alimentazione al circuito: occorre quindi dissipare potenza

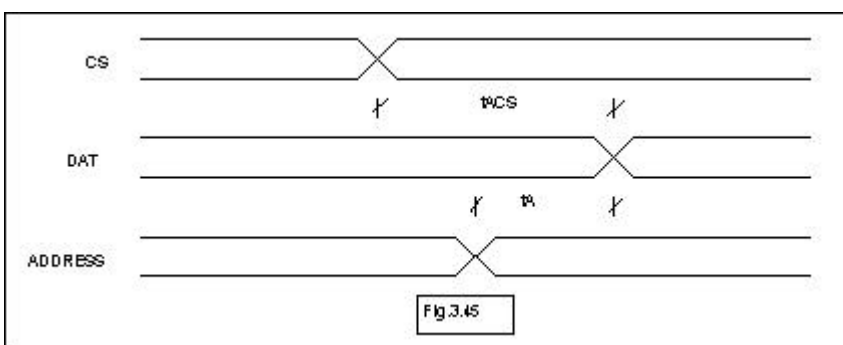
per mantenere immagazzinata l'informazione.



Sono commercialmente disponibili RAM "statiche" e "dinamiche": nelle prime l'elemento di memoria è un FF, nel quale l'informazione (cioè il valore del bit) può permanere un tempo indefinito (a patto, naturalmente, che ci sia alimentazione).

In quelle dinamiche, l'elemento di memoria è un

condensatore, **Fig. 3.43**, e i due stati del bit scritto in esso saranno rappresentati da due diverse quantità di carica immagazzinata. Poiché il condensatore non può conservare la propria carica indefinitamente (la resistenza degli switches aperti è grande, ma non infinita), è necessario "rinfrescare" di tanto in tanto (tipicamente, ogni 2 msec) il contenuto



nitamente (la resistenza degli switches aperti è grande, ma non infinita), è necessario "rinfrescare" di tanto in tanto (tipicamente, ogni 2 msec) il contenuto

della memoria, cioè rifornire al condensatore la carica che ha perso nel frattempo, **Fig. 3.44**.

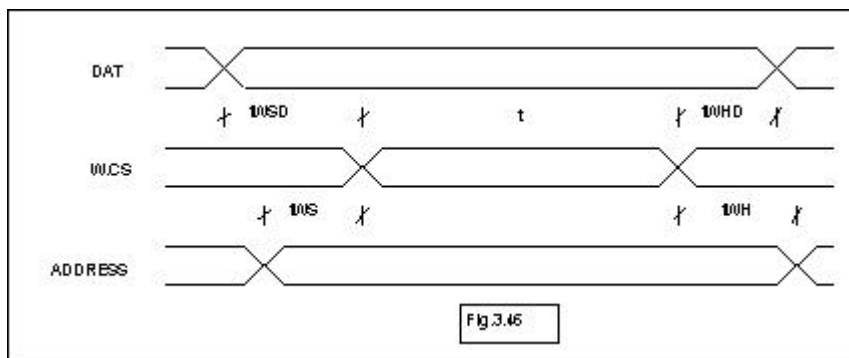
Nelle RAM dinamiche la circuiteria per ogni bit è molto più semplice che in quelle statiche, per cui il

numero di celle di memoria per unità di superficie del chip di silicio è molto maggiore. È invece più complessa la circuiteria di gestione perché deve espletare la funzione di refreshing, non richiesta nelle RAM statiche.

Per le RAM si definiscono dei tempi caratteristici relativi alla velocità di accesso alla memoria; alcuni di essi sono qui descritti.

Per la fase di lettura si definisce il tempo di accesso alla lettura,  $t_{AA}$ : dal momento in cui l'indirizzo è stabile, occorre attendere  $t_{AA}$  secondi prima che l'uscita della memoria sia garantita esatta. Se le parole della memoria sono distribuite su più integrati (questo accade se il numero totale di parole da registrare non può essere contenuto in un solo integrato), è anche presente un segnale di "chip select", il quale individua il circuito integrato (mentre l'indirizzo individua la parola, senza distinguere fra i vari circuiti integrati). Il chip select di norma precede l'indirizzo, per cui il tempo di accesso alla lettura,  $t_{ACS}$ , va riferito a questo segnale. **Fig. 3.45.**

Alla fine della fase di lettura, si definisce un tempo di recupero dopo la lettura,  $t_{RCS}$ : cioè, l'uscita è garantita disabilitata  $t_{RCS}$  secondi dopo la scomparsa del chip select. Questo è il tempo che occorre attendere prima di leggere il contenuto dello stesso indirizzo su un altro integrato, poiché le linee di uscita sono comuni a tutti gli integrati che costituiscono la memoria.



In fase di scrittura, **Fig. 3.46**, perché si possa scrivere un dato devono essere presenti W e CS; tale presenza deve durare un tempo minimo,  $t_w$ , che

garantisce la scrittura corretta. L'indirizzo ed il dato devono essere ovviamente presenti durante  $t_w$ . Più precisamente, l'indirizzo deve essere presente già da  $t_{WSA}$  secondi, prima che inizi la scrittura, e deve perdurare  $t_{WHA}$  secondi dopo la scomparsa di W.CS: questo garantisce che non vengano disturbate altre locazioni di memoria. Per quel che riguarda il dato, i tempi corrispondenti sono  $t_{WSD}$  e  $t_{WHD}$  i quali, insieme con  $t_w$ , garantiscono la scrittura corretta del dato stesso.